Adaptive Semi-supervised Learning for Cross-domain Sentiment Classification

Ruidan He[†‡], **Wee Sun Lee**[†], **Hwee Tou Ng**[†], and **Daniel Dahlmeier**[‡]
[†]Department of Computer Science, National University of Singapore
[‡]SAP Innovation Center Singapore
[†]{ruidanhe,leews,nght}@comp.nus.edu.sg
[‡]d.dahlmeier@sap.com

Abstract

We consider the cross-domain sentiment classification problem, where a sentiment classifier is to be learned from a source domain and to be generalized to a target domain. Our approach explicitly minimizes the distance between the source and the target instances in an embedded feature space. With the difference between source and target minimized, we then exploit additional information from the target domain by consolidating the idea of semi-supervised learning, for which, we jointly employ two regularizations – entropy minimization and self-ensemble bootstrapping – to incorporate the unlabeled target data for classifier refinement. Our experimental results demonstrate that the proposed approach can better leverage unlabeled data from the target domain and achieve substantial improvements over baseline methods in various experimental settings.

1 Introduction

In practice, it is often difficult and costly to annotate sufficient training data for diverse application domains on-the-fly. We may have sufficient labeled data in an existing domain (called the source domain), but very few or no labeled data in a new domain (called the target domain). This issue has motivated research on cross-domain sentiment classification, where knowledge in the source domain is transferred to the target domain in order to alleviate the required labeling effort.

One key challenge of domain adaptation is that data in the source and target domains are drawn from different distributions. Thus, adaptation performance will decline with an increase in distribution difference. Specifically, in sentiment analysis, reviews of different products have different vocabulary. For instance, restaurants reviews would contain opinion words such as "tender", "tasty", or "undercooked" and movie reviews would contain "thrilling", "horrific", or "hilarious". The intersection between these two sets of opinion words could be small which makes domain adaptation difficult.

Several techniques have been proposed for addressing the problem of domain shifting. The aim is to bridge the source and target domains by learning domain-invariant feature representations so that a classifier trained on a source domain can be adapted to another target domain. In cross-domain sentiment classification, many works (Blitzer et al., 2007; Pan et al., 2010; Zhou et al., 2015; Wu and Huang, 2016; Yu and Jiang, 2016) utilize a key intuition that domain-specific features could be aligned with the help of domain-invariant features (pivot features). For instance, "hilarious" and "tasty" could be aligned as both of them are relevant to "good".

Despite their promising results, these works share two major limitations. First, they highly depend on the heuristic selection of pivot features, which may be sensitive to different applications. Thus the learned new representations may not effectively reduce the domain difference. Furthermore, these works only utilize the unlabeled target data for representation learning while the sentiment classifier was solely trained on the source domain. There have not been many studies on exploiting unlabeled target data for refining the classifier, even though it may contain beneficial information. How to effectively leverage unlabeled target data still remains an important challenge for domain adaptation.

In this work, we argue that the information from unlabeled target data is beneficial for domain adaptation and we propose a novel **D**omain **A**daptive **S**emi-supervised learning framework (DAS) to better exploit it. Our main intuition is to treat the problem as a semi-supervised learning task by considering target instances as unla-

Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 3467–3476
Brussels, Belgium, October 31 - November 4, 2018.

beled data, assuming the domain distance can be effectively reduced through domain-invariant representation learning. Specifically, the proposed approach jointly performs feature adaptation and semi-supervised learning in a multi-task learning setting. For feature adaptation, it explicitly minimizes the distance between the encoded representations of the two domains. On this basis, two semi-supervised regularizations – entropy minimization and self-ensemble bootstrapping – are jointly employed to exploit unlabeled target data for classifier refinement.

We evaluate our method rigorously under multiple experimental settings by taking label distribution and corpus size into consideration. The results show that our model is able to obtain significant improvements over strong baselines. We also demonstrate through a series of analysis that the proposed method benefits greatly from incorporating unlabeled target data via semi-supervised learning, which is consistent with our motivation. Our datasets and source code can be obtained from https://github.com/ruidan/DAS.

2 Related Work

Domain Adaptation: The majority of feature adaptation methods for sentiment analysis rely on a key intuition that even though certain opinion words are completely distinct for each domain, they can be aligned if they have high correlation with some domain-invariant opinion words (pivot words) such as "excellent" or "terrible". Blitzer et al. (2007) proposed a method based on structural correspondence learning (SCL), which uses pivot feature prediction to induce a projected feature space that works well for both the source and the target domains. The pivot words are selected in a way to cover common domain-invariant opinion words. Subsequent research aims to better align the domain-specific words (Pan et al., 2010; He et al., 2011; Wu and Huang, 2016) such that the domain discrepancy could be reduced. More recently, Yu and Jiang (2016) borrow the idea of pivot feature prediction from SCL and extend it to a neural network-based solution with auxiliary tasks. In their experiment, substantial improvement over SCL has been observed due to the use of real-valued word embeddings. Unsupervised representation learning with deep neural networks (DNN) such as denoising autoencoders has also been explored for feature adaptation (Glorot et al., 2011; Chen et al., 2012; Yang and Eisenstein, 2014). It has been shown that DNNs could learn transferable representations that disentangle the underlying factors of variation behind data samples.

Although the aforementioned methods aim to reduce the domain discrepancy, they do not explicitly minimize the distance between distributions, and some of them highly rely on the selection of pivot features. In our method, we formally construct an objective for this purpose. Similar ideas have been explored in many computer vision problems, where the representations of the underlying domains are encouraged to be similar through explicit objectives (Tzeng et al., 2014; Ganin and Lempitsky, 2015; Long et al., 2015; Zhuang et al., 2015; Long et al., 2017) such as maximum mean discrepancy (MMD) (Gretton et al., 2012). In NLP tasks, Li et al. (2017) and Chen et al. (2017) both proposed using adversarial training framework for reducing domain difference. In their model, a subnetwork is added as a domain discriminator while deep features are learned to confuse the discriminator. The feature adaptation component in our model shares similar intuition with MMD and adversary training. We will show a detailed comparison with them in our experiments.

Semi-supervised Learning: We attempt to treat domain adaptation as a semi-supervised learning task by considering the target instances as unlabeled data. Some efforts have been initiated on transfer learning from unlabeled data (Dai et al., 2007; Jiang and Zhai, 2007; Wu et al., 2009). In our model, we reduce the domain discrepancy by feature adaptation, and thereafter adopt semi-supervised learning techniques to learn from unlabeled data. Primarily motivated by (Grandvalet and Bengio, 2004) and (Laine and Aila, 2017), we employed entropy minimization and self-ensemble bootstrapping as regularizations to incorporate unlabeled data. Our experimental results show that both methods are effective when jointly trained with the feature adaptation objective, which confirms to our motivation.

3 Model Description

3.1 Notations and Model Overview

We conduct most of our experiments under an unsupervised domain adaptation setting, where we have no labeled data from the target domain. Consider two sets D_s and D_t. $D_s = \{\mathbf{x}_i^{(s)}, \mathbf{y}_i^{(s)}\}|_{i=1}^{n_s}$ is

from the source domain with n_s labeled examples, where $\mathbf{y}_i \in \mathbb{R}^C$ is a one-hot vector representation of sentiment label and C denotes the number of classes. $D_t = \{\mathbf{x}_i^{(t)}\}|_{i=1}^{n_t}$ is from the target domain with n_t unlabeled examples. $N = n_s + n_t$ denotes the total number of training documents including both labeled and unlabeled[1]. We aim to learn a sentiment classifier from D_s and D_t such that the classifier would work well on the target domain. We also present some results under a setting where we assume that a small number of labeled target examples are available (see Figure 3).

For the proposed model, we denote G parameterized by θ_g as a neural-based feature encoder that maps documents from both domains to a shared feature space, and $\mathcal{F}$ parameterized by θ_f as a fully connected layer with softmax activation serving as the sentiment classifier. We aim to learn feature representations that are domain-invariant and at the same time discriminative on both domains, thus we simultaneously consider three factors in our objective: (1) minimize the classification error on the labeled source examples; (2) minimize the domain discrepancy; and (3) leverage unlabeled data via semi-supervised learning.

Suppose we already have the encoded features of documents $\{\boldsymbol{\xi}_i^{(s,t)} = G(\mathbf{x}_i^{(s,t)}; \theta_g)\}|_{i=1}^{N}$ (see Section 4.1), the objective function for purpose (1) is thus the cross entropy loss on the labeled source examples

$$L = -\frac{1}{n_s} \sum_{i=1}^{n_s} \sum_{j=1}^{C} \mathbf{y}_i^{(s)}(j) \log \tilde{\mathbf{y}}_i^{(s)}(j) \quad (1)$$

where $\tilde{\mathbf{y}}_i^{(s)} = \mathcal{F}(\boldsymbol{\xi}_i^{(s)}; \theta_f)$ denotes the predicted label distribution. In the following subsections, we will explain how to perform feature adaptation and domain adaptive semi-supervised learning in details for purpose (2) and (3) respectively.

3.2 Feature Adaptation

Unlike prior works (Blitzer et al., 2007; Yu and Jiang, 2016), our method does not attempt to align domain-specific words through pivot words. In our preliminary experiments, we found that word embeddings pre-trained on a large corpus are able to adequately capture this information. As we will later show in our experiments, even without adaptation, a naive neural network classifier with pre-trained word embeddings can already achieve reasonably good results.

We attempt to explicitly minimize the distance between the source and target feature representations ($\{\boldsymbol{\xi}_i^{(s)}\}|_{i=1}^{n_s}$ and $\{\boldsymbol{\xi}_i^{(t)}\}_{i=1}^{n_t}$). A few methods from literature can be applied such as Maximum Mean Discrepancy (MMD) (Gretton et al., 2012) or adversary training (Li et al., 2017; Chen et al., 2017). The main idea of MMD is to estimate the distance between two distributions as the distance between sample means of the projected embeddings in Hilbert space. MMD is implicitly computed through a characteristic kernel, which is used to ensure that the sample mean is injective, leading to the MMD being zero if and only if the distributions are identical. In our implementation, we skip the mapping procedure induced by a characteristic kernel for simplifying the computation and learning. We simply estimate the distribution distance as the distance between the sample means in the current embedding space. Although this approximation cannot preserve all statistical features of the underlying distributions, we find it performs comparably to MMD on our problem. The following equations formally describe the feature adaptation loss $\mathcal{J}$:

$$\mathcal{J} = KL(\mathbf{g}_s||\mathbf{g}_t) + KL(\mathbf{g}_t||\mathbf{g}_s) \quad (2)$$

$$\mathbf{g}'_s = \frac{1}{n_s} \sum_{i=1}^{n_s} \boldsymbol{\xi}_i^{(s)}, \qquad \mathbf{g}_s = \frac{\mathbf{g}'_s}{\|\mathbf{g}'_s\|_1} \quad (3)$$

$$\mathbf{g}'_t = \frac{1}{n_t} \sum_{i=1}^{n_t} \boldsymbol{\xi}_i^{(t)}, \qquad \mathbf{g}_t = \frac{\mathbf{g}'_t}{\|\mathbf{g}'_t\|_1} \quad (4)$$

L_1 normalization is applied on the mean representations $\mathbf{g}'_s$ and $\mathbf{g}'_t$, rescaling the vectors such that all entries sum to 1. We adopt a symmetric version of KL divergence (Zhuang et al., 2015) as the distance function. Given two distribution vectors $P, Q \in \mathbb{R}^k$, $KL(P||Q) = \sum_{i=1}^{k} P(i) \log(\frac{P(i)}{Q(i)})$.

3.3 Domain Adaptive Semi-supervised Learning (DAS)

We attempt to exploit the information in target data through semi-supervised learning objectives, which are jointly trained with L and $\mathcal{J}$. Normally, to incorporate target data, we can minimize the cross entropy loss between the true label distributions $\mathbf{y}_i^{(t)}$ and the predicted label distributions

[1]Note that unlabeled source examples can also be included for training. In that case, $N = n_s + n_t + n_{s'}$ where $n_{s'}$ denotes the number of unlabeled source examples. This corresponds to our experimental setting 2. For simplicity, we only consider n_s and n_t in our description.

$\tilde{\mathbf{y}}_i^{(t)} = \mathcal{F}(\boldsymbol{\xi}_i^{(t)}; \theta_f)$ over target samples. The challenge here is that $\mathbf{y}_i^{(t)}$ is unknown, and thus we attempt to estimate it via semi-supervised learning. We use entropy minimization and bootstrapping for this purpose. We will later show in our experiments that both methods are effective, and jointly employing them overall yields the best results.

Entropy Minimization: In this method, $\mathbf{y}_i^{(t)}$ is estimated as the predicted label distribution $\tilde{\mathbf{y}}_i^{(t)}$, which is a function of θ_g and θ_f. The loss can thus be written as

$$\Gamma = -\frac{1}{n_t} \sum_{i=1}^{n_t} \sum_{j=1}^{C} \tilde{\mathbf{y}}_i^{(t)}(j) \log \tilde{\mathbf{y}}_i^{(t)}(j) \qquad (5)$$

Assume the domain discrepancy can be effectively reduced through feature adaptation, by minimizing the entropy penalty, training of the classifier is influenced by the unlabeled target data and will generally maximize the margins between the target examples and the decision boundaries, increasing the prediction confidence on the target domain.

Self-ensemble Bootstrapping: Another way to estimate $\mathbf{y}_i^{(t)}$ corresponds to bootstrapping. The idea is to estimate the unknown labels as the predictions of the model learned from the previous round of training. Bootstrapping has been explored for domain adaptation in previous works (Jiang and Zhai, 2007; Wu et al., 2009). However, in their methods, domain discrepancy was not explicitly minimized via feature adaptation. Applying bootstrapping or other semi-supervised learning techniques in this case may worsen the results as the classifier can perform quite bad on the target data.

Inspired by the ensembling method proposed in (Laine and Aila, 2017), we estimate $\mathbf{y}_i^{(t)}$ by forming ensemble predictions of labels during training, using the outputs on different training epochs. The loss is formulated as follows:

$$\Omega = -\frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{C} \tilde{\mathbf{z}}_i^{(s,t)}(j) \log \tilde{\mathbf{y}}_i^{(s,t)}(j) \qquad (6)$$

where $\tilde{\mathbf{z}}$ denotes the estimated labels computed on the ensemble predictions from different epochs. The loss is applied on all documents. It serves for bootstrapping on the unlabeled target data, and it also serves as a regularization that encourages the network predictions to be consistent in different training epochs. Ω is jointly trained with L, $\mathcal{J}$, and Γ. Algorithm 1 illustrates the overall training process of the proposed domain adaptive semi-supervised learning (DAS) framework.

Algorithm 1 Pseudocode for training DAS

Require: D_s, D_t, G, $\mathcal{F}$
Require: α = ensembling momentum, $0 \leq \alpha < 1$
Require: $w(t)$ = weight ramp-up function
  $\mathbf{Z} \leftarrow 0_{[N \times C]}$
  $\tilde{\mathbf{z}} \leftarrow 0_{[N \times C]}$
  for $t \in [1, \textit{max-epochs}]$ **do**
    for each minibatch $B^{(s)}$, $B^{(t)}$, $B^{(u)}$ in D_s, D_t, $\{x_i^{(s,t)}\}|_{i=1}^{N}$ **do**
      compute loss L on $[\mathbf{x}_{i \in B^{(s)}}, \mathbf{y}_{i \in B^{(s)}}]$
      compute loss $\mathcal{J}$ on $[\mathbf{x}_{i \in B^{(s)}}, \mathbf{x}_{j \in B^{(t)}}]$
      compute loss Γ on $\mathbf{x}_{i \in B^{(t)}}$
      compute loss Ω on $[\mathbf{x}_{i \in B^{(u)}}, \tilde{\mathbf{z}}_{i \in B^{(u)}}]$
      overall-loss $\leftarrow L + \lambda_1 \mathcal{J} + \lambda_2 \Gamma + w(t)\Omega$
      update network parameters
    end for
    $\mathbf{Z}'_i \leftarrow \mathcal{F}(G(\mathbf{x}_i))$, for $i \in N$
    $\mathbf{Z} \leftarrow \alpha \mathbf{Z} + (1 - \alpha)\mathbf{Z}'$
    $\tilde{\mathbf{z}} \leftarrow \textit{one-hot-vectors}(\mathbf{Z})$
  end for

In Algorithm 1, λ_1, λ_2, and $w(t)$ are weights to balance the effects of $\mathcal{J}$, Γ, and Ω respectively. λ_1 and λ_2 are constant hyper-parameters. We set $w(t) = \exp[-5(1 - \frac{t}{\textit{max-epochs}})^2]\lambda_3$ as a Gaussian curve to ramp up the weight from 0 to λ_3. This is to ensure the ramp-up of the bootstrapping loss component is slow enough in the beginning of the training. After each training epoch, we compute $\mathbf{Z}'_i$ which denotes the predictions made by the network in current epoch, and then the ensemble prediction $\mathbf{Z}_i$ is updated as a weighted average of the outputs from previous epochs and the current epoch, with recent epochs having larger weight. For generating estimated labels $\tilde{\mathbf{z}}_i$, $\mathbf{Z}_i$ is converted to a one-hot vector where the entry with the maximum value is set to one and other entries are set to zeros. The self-ensemble bootstrapping is a generalized version of bootstrappings that only use the outputs from the previous round of training (Jiang and Zhai, 2007; Wu et al., 2009). The ensemble prediction is likely to be closer to the correct, unknown labels of the target data.

Domain		#Pos	#Neg	#Neu	Total
Book	Set 1	2000	2000	2000	6000
	Set 2	4824	513	663	6000
Electronics	Set 1	2000	2000	2000	6000
	Set 2	4817	694	489	6000
Beauty	Set 1	2000	2000	2000	6000
	Set 2	4709	616	675	6000
Music	Set 1	2000	2000	2000	6000
	Set 2	4441	785	774	6000

(a) Small-scale datasets

Domain	#Pos	#Neg	#Neu	Total
IMDB	55,242	11,735	17,942	84,919
Yelp	155,625	29,597	45,941	231,163
Cell Phone	148,657	24,343	21,439	194,439
Baby	126,525	17,012	17,255	160,792

(b) Large-scale datasets

Table 1: Summary of datasets.

4 Experiments

4.1 CNN Encoder Implementation

We have left the feature encoder G unspecified, for which, a few options can be considered. In our implementation, we adopt a one-layer CNN structure from previous works (Kim, 2014; Yu and Jiang, 2016), as it has been demonstrated to work well for sentiment classification tasks. Given a review document $\mathbf{x} = (x_1, x_2, ..., x_n)$ consisting of n words, we begin by associating each word with a continuous word embedding (Mikolov et al., 2013) $\mathbf{e}_x$ from an embedding matrix $\mathbf{E} \in \mathbb{R}^{V \times d}$, where V is the vocabulary size and d is the embedding dimension. E is jointly updated with other network parameters during training. Given a window of dense word embeddings $\mathbf{e}_{x_1}, \mathbf{e}_{x_2}, ..., \mathbf{e}_{x_l}$, the convolution layer first concatenates these vectors to form a vector $\hat{\mathbf{x}}$ of length ld and then the output vector is computed by Equation (7):

$$Conv(\hat{\mathbf{x}}) = f(\mathbf{W} \cdot \hat{\mathbf{x}} + \mathbf{b}) \tag{7}$$

$\theta_g = \{\mathbf{W}, \mathbf{b}\}$ is the parameter set of the encoder G and is shared across all windows of the sequence. f is an element-wise non-linear activation function. The convolution operation can capture local contextual dependencies of the input sequence and the extracted feature vectors are similar to n-grams. After the convolution operation is applied to the whole sequence, we obtain a list of hidden vectors $\mathbf{H} = (\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_n)$. A max-over-time pooling layer is applied to obtain the final vector representation $\boldsymbol{\xi}$ of the input document.

4.2 Datasets and Experimental Settings

Existing benchmark datasets such as the Amazon benchmark (Blitzer et al., 2007) typically remove reviews with neutral labels in both domains. This is problematic as the label information of the target domain is not accessible in an unsupervised domain adaptation setting. Furthermore, removing neutral instances may bias the dataset favorably for max-margin-based algorithms like ours, since the resulting dataset has all uncertain labels removed, leaving only high confidence examples. Therefore, we construct new datasets by ourselves. The results on the original Amazon benchmark is qualitatively similar, and we present them in Appendix A for completeness since most of previous works reported results on it.

Small-scale datasets: Our new dataset was derived from the large-scale Amazon datasets[2] released by McAuley et al. (2015). It contains four domains[3]: Book (BK), Electronics (E), Beauty (BT), and Music (M). Each domain contains two datasets. Set 1 contains 6000 instances with exactly balanced class labels, and set 2 contains 6000 instances that are randomly sampled from the large dataset, preserving the original label distribution, which we believe better reflects the label distribution in real life. The examples in these two sets do not overlap. Detailed statistics of the generated datasets are given in Table 1a.

In all our experiments on the small-scale datasets, we use set 1 of the source domain as the only source with sentiment label information during training, and we evaluate the trained model on set 1 of the target domain. Since we cannot control the label distribution of unlabeled data during training, we consider two different settings:

Setting (1): Only set 1 of the target domain is used as the unlabeled set. This tells us how the method performs in a condition when the target domain has a close-to-balanced label distribution. As we also evaluate on set 1 of the target domain, this is also considered as a transductive setting.

Setting (2): Set 2 from both the source and target domains are used as unlabeled sets. Since set 2 is directly sampled from millions of reviews, it better reflects real-life sentiment distribution.

Large-scale datasets: We further conduct experiments on four much larger datasets: IMDB[4]

[2] http://jmcauley.ucsd.edu/data/amazon/

[3] The original reviews were rated on a 5-point scale. We label them with rating < 3, > 3, and $= 3$ as negative, positive, and neutral respectively.

[4] IMDB is rated on a 10-point scale, and we label reviews with rating < 5, > 6, and $= 5/6$ as negative, positive, and neutral respectively.

(I), Yelp2014 (Y), Cell Phone (C), and Baby (B). IMDB and Yelp2014 were previously used in (Tang et al., 2015; Yang et al., 2017). Cell phone and Baby are from the large-scale Amazon dataset (McAuley et al., 2015; He and McAuley, 2016). Detailed statistics are summarized in Table 1b. We keep all reviews in the original datasets and consider a transductive setting where all target examples are used for both training (without label information) and evaluation. We perform sampling to balance the classes of labeled source data in each minibatch $B^{(s)}$ during training.

4.3 Selection of Development Set

Ideally, the development set should be drawn from the same distribution as the test set. However, under the unsupervised domain adaptation setting, we do not have any labeled target data at training phase which could be used as development set. In all of our experiments, for each pair of domains, we instead sample 1000 examples from the training set of the source domain as development set. We train the network for a fixed number of epochs, and the model with the minimum classification error on this development set is saved for evaluation. This approach works well on most of the problems since the target domain is supposed to behave like the source domain if the domain difference is effectively reduced.

Another problem is how to select the values for hyper-parameters. If we tune λ_1 and λ_2 directly on the development set from the source domain, most likely both of them will be set to 0, as unlabeled target data is not helpful for improving in-domain accuracy of the source domain. Other neural network models also have the same problem for hyper-parameter tuning. Therefore, our strategy is to use the development set from the target domain to optimize λ_1 and λ_2 for one problem (e.g., we only do this on E→BK), and fix their values on the other problems. This setting assumes that we have at least two labeled domains such that we can optimize the hyper-parameters, and then we fix them for other new unlabeled domains to transfer to.

4.4 Training Details and Hyper-parameters

We initialize word embeddings using the 300-dimension GloVe vectors supplied by Pennington et al., (2014), which were trained on 840 billion tokens from the Common Crawl. For each pair of domains, the vocabulary consists of the top 10000 most frequent words. For words in the vocabulary but not present in the pre-trained embeddings, we randomly initialize them.

We set hyper-parameters of the CNN encoder following previous works (Kim, 2014; Yu and Jiang, 2016) without specific tuning on our datasets. The window size is set to 3 and the size of the hidden layer is set to 300. The nonlinear activation function is Relu. For regularization, we also follow their settings and employ dropout with probability set to 0.5 on $\boldsymbol{\xi}_i$ before feeding it to the output layer $\mathcal{F}$, and constrain the l_2-norm of the weight vector θ_f, setting its max norm to 3.

On the small-scale datasets and the Aamzon benchmark, λ_1 and λ_2 are set to 200 and 1, respectively, tuned on the development set of task E→BK under setting 1. On the large-scale datasets, λ_1 and λ_2 are set to 500 and 0.2, respectively, tuned on I→Y. We use a Gaussian curve $w(t) = \exp[-5(1 - \frac{t}{t_{max}})^2]\lambda_3$ to ramp up the weight of the bootstrapping loss Ω from 0 to λ_3, where t_{max} denotes the maximum number of training epochs. We train 30 epochs for all experiments. We set λ_3 to 3 and α to 0.5 for all experiments.

The batch size is set to 50 on the small-scale datasets and the Amazon benchmark. We increase the batch size to 250 on the large-scale datasets to reduce the number of iterations. RMSProp optimizer with learning rate set to 0.0005 is used for all experiments.

4.5 Models for Comparison

We compare with the following baselines:

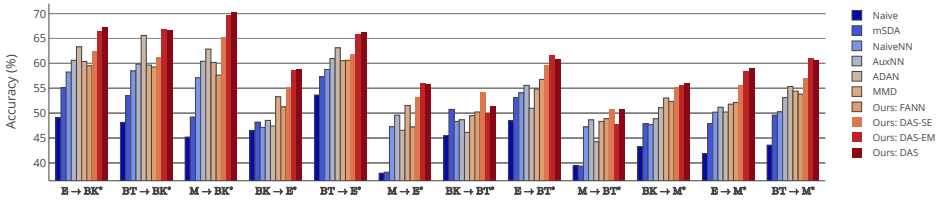
(1) **Naive**: A non-domain-adaptive baseline with bag-of-words representations and SVM classifier trained on the source domain.

(2) **mSDA** (Chen et al., 2012): This is the state-of-the-art method based on discrete input features. Top 1000 bag-of-words features are kept as pivot features. We set the number of stacked layers to 3 and the corruption probability to 0.5.

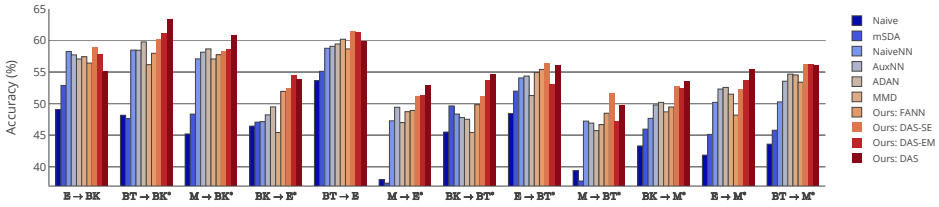
(3) **NaiveNN**: This is a non-domain-adaptive CNN trained on source domain, which is a variant of our model by setting λ_1, λ_2, and λ_3 to zeros.

(4) **AuxNN** (Yu and Jiang, 2016): This is a neural model that exploits auxiliary tasks, which has achieved state-of-the-art results on cross-domain sentiment classification. The sentence encoder used in this model is the same as ours.

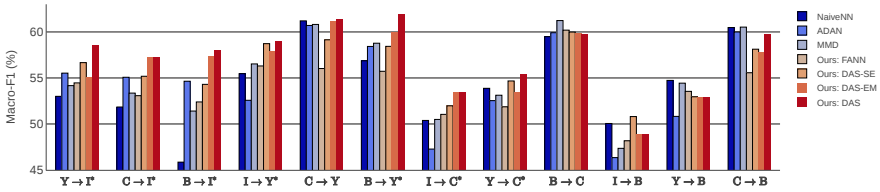
(5) **ADAN** (Chen et al., 2017): This method exploits adversarial training to reduce representa-



(a) Accuracy on the small-scale dataset under setting 1.



(b) Accuracy on the small-scale dataset under setting 2.



(c) Macro-F1 on the large-scale dataset.

Figure 1: Performance comparison. Average results over 5 runs with random initializations are reported for each neural method. * indicates that the proposed method (either of DAS, DAS-EM, DAS-SE) is significantly better than other baselines (baseline 1-6) with $p < 0.05$ based on one-tailed unpaired t-test.

tion difference between domains. The original paper uses a simple feedforward network as encoder. For fair comparison, we replace it with our CNN-based encoder. We train 5 iterations on the discriminator per iteration on the encoder and sentiment classifier as suggested in their paper.

(6) **MMD**: MMD has been widely used for minimizing domain discrepancy on images. In those works (Tzeng et al., 2014; Long et al., 2017), variants of deep CNNs are used for encoding images and the MMDs of multiple layers are jointly minimized. In NLP, adding more layers of CNNs may not be very helpful and thus those models from image-related tasks can not be directly applied to our problem. To compare with MMD-based method, we train a model that jointly minimize the classification loss L on the source domain and MMD between $\{\boldsymbol{\xi}_i^{(s)}|_{i=1}^{n_s}\}$ and $\{\boldsymbol{\xi}_i^{(t)}|_{i=1}^{n_t}\}$. For computing MMD, we use a Gaussian RBF which is a common choice for characteristic kernel.

In addition to the above baselines, we also show results of different variants of our model. **DAS** as shown in Algorithm 1 denotes our full model. **DAS-EM** denotes the model with only entropy minimization for semi-supervised learning (set $\lambda_3 = 0$). **DAS-SE** denotes the model with only self-ensemble bootstrapping for semi-supervised learning (set $\lambda_2 = 0$). **FANN** (feature-adaptation neural network) denotes the model without semi-supervised learning performed (set both λ_2 and λ_3 to zeros).

4.6 Main Results

Figure 1[5] shows the comparison of adaptation results (see Appendix B for the exact numerical numbers). We report classification accuracy on the small-scale dataset. For the large-scale dataset, macro-F1 is instead used since the label distribution in the test set is extremely unbalanced. Key observations are summarized as follows. (1) Both DAS-EM and DAS-SE perform better in most cases compared with ADAN, MDD, and FANN, in which only feature adaptation is performed. This demonstrates the effectiveness of the pro-

[5]We exclude results of Naive, mSDA and AuxNN on the large-scale dataset. Both Naive and mSDA have difficulties to scale up to the large dataset. AuxNN relies on manually selecting positive and negative pivots before training.

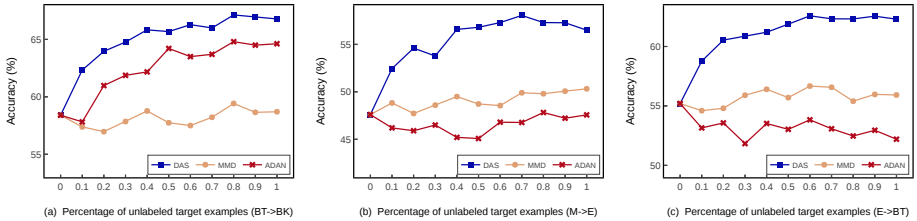


Figure 2: Accuracy vs. percentage of unlabeled target training examples.

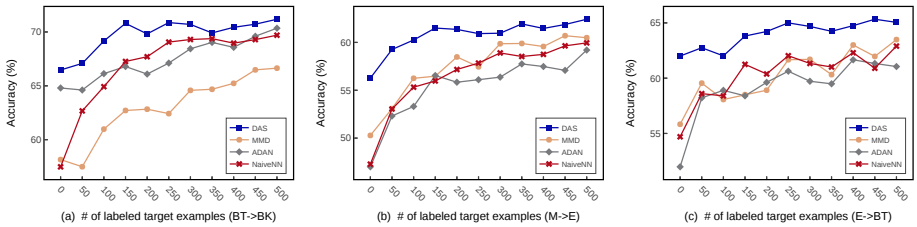


Figure 3: Accuracy vs. number of labeled target training examples.

posed domain adaptive semi-supervised learning framework. DAS-EM is more effective than DAS-SE in most cases, and the full model DAS with both techniques jointly employed overall has the best performance. (2) When comparing the two settings on the small-scale dataset, all domain-adaptive methods[6] generally perform better under setting 1. In setting 1, the target examples are balanced in classes, which can provide more diverse opinion-related features. However, when considering unsupervised domain adaptation, we should not presume the label distribution of the unlabeled data. Thus, it is necessary to conduct experiments using datasets that reflect real-life sentiment distribution as what we did on setting2 and the large-scale dataset. Unfortunately, this is ignored by most of previous works. (3) Word-embeddings are very helpful, as we can see even NaiveNN can substantially outperform mSDA on most tasks.

To see the effect of semi-supervised learning alone, we also conduct experiments by setting $\lambda_1 = 0$ to eliminate the effect of feature adaptation. Both entropy minimization and bootstrapping perform very badly in this setting. Entropy minimization gives almost random predictions with accuracy below 0.4, and the results of bootstrapping are also much lower compared to NaiveNN. This suggests that the feature adaptation component is essential. Without it, the learned target representations are less meaningful and discriminative. Applying semi-supervised learning in this case is likely to worsen the results.

4.7 Further Analysis

In Figure 2, we show the change of accuracy with respect to the percentage of unlabeled data used for training on three particular problems under setting 1. The value at $x = 0$ denotes the accuracies of NaiveNN which does not utilize any target data. For DAS, we observe a nonlinear increasing trend where the accuracy quickly improves at the beginning, and then gradually stabilizes. For other methods, this trend is less obvious, and adding more unlabeled data sometimes even worsen the results. This finding again suggests that the proposed approach can better exploit the information from unlabeled data.

We also conduct experiments under a setting with a small number of labeled target examples available. Figure 3 shows the change of accuracy with respect to the number of labeled target examples added for training. We can observe that DAS is still more effective under this setting, while the performance differences to other methods gradually decrease with the increasing number of labeled target examples.

4.8 CNN Filter Analysis

In this subsection, we aim to better understand DAS by analyzing sentiment-related CNN filters. To do that, 1) we first select a list of the most related CNN filters for predicting each sentiment label (positive, negative neutral). Those filters can be identified according to the learned weights θ_f

[6]Results of Naive and NaiveNN do not change under both settings as they are only trained on the source domain.

best-value-at	highly-recommend-!	nars-are-amazing	beauty-store-suggested	since-i-love
good-value-at	highly-advise-!	ulta-are-fantastic	durable-machine-and	years-i-love
perfect-product-for	gogeous-absolutely-perfect	length-are-so	perfect-length-and	bonus-i-love
great-product-at	love-love-love	expected-in-perfect	great-store-on	appearance-i-love
amazing-product-∗	highly-recommend-for	setting-works-perfect	beauty-store-for	relaxing-i-love

(a) NaiveNN

prices-my-favorite	so-nicely-!	purchase-thanks-!	feel-wonderfully-clean	are-really-cleaning
brands-my-favorite	more-affordable-price	buy-again-!	on-nicely-builds	washing-and-cleaning
very-great-stores	shampoo-a-perfect	without-hesitation-!	polish-easy-and	really-good-shampoo
great-bottle-also	an-excellent-value	buy-this-!	felt-cleanser-than	deeply-cleans-my
scent-pleasantly-floral	really-enjoy-it	discount-too-!	honestly-perfect-it	totally-moisturize-our

(b) FANN

bath-'s-wonderful	love-fruity-sweet	feeling-smooth-radiant	cleans-thoroughly-*	excellent-everyday-lotion
all-pretty-affordable	absorb-really-nicely	love-lavender-scented	loving-this-soap	affordable-cleans-nicely
it-delivers-fabulous	shower-lather-wonderfully	am-very-grateful	bed-of-love	fantastic-base-coat
and-blends-nicely	*-smells-fantastic	love-fruity-fragrances	shower-!-*	nice-gentle-scrub
heats-quickly-love	and-clean-excellent	perfect-beautiful-shimmer	radiant-daily-moisturizer	surprisingly-safe-on

(c) DAS

Table 2: Comparison of the top trigrams (each column) from the target domain (beauty) captured by the 5 most positive-sentiment-related CNN filters learned on E→BT. $*$ denotes a padding.

of the output layer $\mathcal{F}$. Higher weight indicates stronger relatedness. 2) Recall that in our implementation, each CNN filter has a window size of 3 with Relu activation. We can thus represent each selected filter as a ranked list of trigrams with highest activation values.

We analyze the CNN filters learned by NaiveNN, FANN and DAS respectively on task E→BT under setting 1. We focus on E→BT for study because electronics and beauty are very different domains and each of them has a diverse set of domain-specific sentiment expressions. For each method, we identify the top 10 most related filters for each sentiment label, and extract the top trigrams of each selected filter on both source and target domains. Since labeled source examples are used for training, we find the filters learned by the three methods capture similar expressions on the source domain, containing both domain-invariant and domain-specific trigrams. On the target domain, DAS captures more target-specific expressions compared to the other two methods. Due to space limitation, we only present a small subset of positive-sentiment-related filters in Table 2. The complete results are provided in Appendix C. From Table 2, we can observe that the filters learned by NaiveNN are almost unable to capture target-specific sentiment expressions, while FANN is able to capture limited target-specific words such as "clean" and "scent". The filters learned by DAS are more domain-adaptive, capturing diverse sentiment expressions in the target domain.

5 Conclusion

In this work, we propose DAS, a novel framework that jointly performs feature adaptation and semi-supervised learning. We have demonstrated through multiple experiments that DAS can better leverage unlabeled data, and achieve substantial improvements over baseline methods. We have also shown that feature adaptation is an essential component, without which, semi-supervised learning is not able to function properly. The proposed framework could be potentially adapted to other domain adaptation tasks, which is the focus of our future studies.

References

John Blitzer, Mark Dredze, and Fernando Pereira. 2007. Biographies, Bollywood, boom-boxes and blenders: domain adaptation for sentiment classification. In *Annual Meeting of the Association for Computational Linguistics*.

Minmin Chen, Zhixiang Xu, Kilian Q. Weinberger, and Fei Sha. 2012. Marginalized denoising autoencoders for domain adaptation. In *The 29th International Conference on Machine Learning*.

Xilun Chen, Yu Sun, Ben Athiwarakun, Claire Cardie, and Kilian Weinberger. 2017. Adversarial deep averaging networks for cross-lingual sentiment classifier. In *Arxiv e-prints arXiv:1606.01614*.

Wenyuan Dai, Gui rong Xue, Qiang Yang, and Yong Yu. 2007. Transferring naive Bayes classifiers for text classification. In *AAAI Conference on Artificial Intelligence*.

Yaroslav Ganin and Victor Lempitsky. 2015. Unsupervised domain adaptation by backpropagation. In *International Conference on Machine Learning*.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Domain adaptation for large-scale sentiment classification: a deep learning approach. In *The 28th International Conference on Machine Learning*.

Yves Grandvalet and Yoshua Bengio. 2004. Semi-supervised learning by entropy minimization. In *Neural Information Processing Systems*.

Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. 2012. A kernel two-sample test. *Journal of Machine Learning Research*, 13:723–773.

Ruining He and Julian McAuley. 2016. Ups and downs: modeling the visual evolution of fashion trends with one-class collaborative filtering. In *WWW*.

Yulan He, Chenghua Lin, and Harith Alani. 2011. Automatically extracting polarity-bearing topics for cross-domain sentiment classification. In *ACL*.

Jing Jiang and ChengXiang Zhai. 2007. Instance weighting for domain adaptation in NLP. In *Annual Meeting of the Association for Computational Linguistics*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Conference on Empirical Methods in Natural Language Processing*.

Samuli Laine and Timo Aila. 2017. Temporal ensembling for semi-supervised learning. In *International Conference on Learning Representation*.

Zheng Li, Yun Zhang, Ying Wei, Yuxiang Wu, and Qiang Yang. 2017. End-to-end adversarial memory network for cross-domain sentiment classification. In *The 26th International Joint Conference on Artificial Intelligence*.

Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I. Jordan. 2015. Learning transferable features with deep adaptation networks. In *International Conference on Machine Learning*.

Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I. Jordan. 2017. Deep transfer learning with joint adaptation networks. In *International Conference on Machine Learning*.

Julian J. McAuley, Christopher Targett, Qinfeng Shi, and Anton van den Hengel. 2015. Image-based recommendations on styles and substitutes. In *The 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Neural Information Processing Systems*.

Sinno Jialin Pan, Xiaochuan Ni, Jian-Tao Sun, Qiang Yang, and Zheng Chen. 2010. Cross-domain sentiment classification via spectral feature alignment. In *The 19th International World Wide Web Conference*.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. GloVe: Global vectors for word representation. In *Conference on Empirical Methods in Natural Language Processing*.

Duyu Tang, Bing Qin, and Ting Liu. 2015. Learning semantic representation of users and products for document level sentiment classification. In *Annual Meeting of the Association for Computational Linguistics*.

Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. 2014. Deep domain confusion: maximizing for domain invariance. In *Arxiv e-prints arXiv:1412.3474*.

Dan Wu, Wee Sun Lee, Nan Ye, and Hai Leong Chieu. 2009. Domain adaptive bootstrapping for named entity recognition. In *Conference on Empirical Methods in Natural Language Processing*.

Fangzhao Wu and Yongfeng Huang. 2016. Sentiment domain adaptation with multiple sources. In *Annual Meeting of the Association for Computational Linguistics*.

Wei Yang, Wei Lu, and Vincent W. Zheng. 2017. A simple regularization-based algorithm for learning cross-domain word embeddings. In *Conference on Empirical Methods in Natural Language Processing*.

Yi Yang and Jacob Eisenstein. 2014. Fast easy unsupervised domain adaptation with marginalized structured dropout. In *Annual Meeting of the Association for Computational Linguistics*.

Jianfei Yu and Jing Jiang. 2016. Learning sentence embeddings with auxiliary tasks for cross-domain sentiment classification. In *Conference on Empirical Methods in Natural Language Processing*.

Guangyou Zhou, Tingting He, Wensheng Wu, and Xiaohua Tony Hu. 2015. Linking heterogeneous input features with pivots for domain adaptation. In *The 24th International Joint Conference on Artificial Intelligence*.

Guangyou Zhou, Zhiwen Xie, Jimmy Xiangji Huang, and Tingting He. 2016. Bi-transferring deep neural networks for domain adaptation. In *Annual Meeting of the Association for Computational Linguistics*.

Fuzhen Zhuang, Xiaohu Cheng, Ping Luo, Sinno Jialin Pan, and Qing He. 2015. Supervised representation learning: transfer learning with deep autoencoders. In *The 24th International Joint Conference on Artificial Intelligence*.