Wasserstein Selective Transfer Learning for Cross-domain Text Mining

Lingyun Feng[1]*, **Minghui Qiu**[2]*, **Yaliang Li, Hai-Tao Zheng**[1]†, **Ying Shen**[2]†
[1] Tsinghua University, [2] Alibaba Group, [3] Sun-Yat Sen University
fly19@mails.tsinghua.edu.cn,
{minghui.qmh,yaliang.li}@alibaba-inc.com,
sheny76@mail.sysu.edu.cn,zheng.haitao@sz.tsinghua.edu.cn

Abstract

Transfer learning (TL) seeks to improve the learning of a data-scarce target domain by using information from source domains. However, the source and target domains usually have different data distributions, which may lead to negative transfer. To alleviate this issue, we propose a Wasserstein Selective Transfer Learning (WSTL) method. Specifically, the proposed method considers a reinforced selector to select helpful data for transfer learning. We further use a Wasserstein-based discriminator to maximize the empirical distance between the selected source data and target data. The TL module is then trained to minimize the estimated Wasserstein distance in an adversarial manner and provides domain invariant features for the reinforced selector. We adopt an evaluation metric based on the performance of the TL module as delayed reward and a Wasserstein-based metric as immediate rewards to guide the reinforced selector learning. Compared with the competing TL approaches, the proposed method selects data samples that are closer to the target domain. It also provides better state features and reward signals that lead to better performance with faster convergence. Extensive experiments on three real-world text mining tasks demonstrate the effectiveness of the proposed method.

1 Introduction

Transfer learning (TL) is a type of classical machine learning methods to leverage information from data-rich source domains to help a data-scarce target domain (Pan and Yang, 2009). Recently, transfer learning based on deep neural networks, referred to as deep transfer learning (Ruder and Plank, 2017; Yosinski et al., 2014), has been widely used on various tasks in natural language processing (Peng et al., 2018; Mou et al., 2016; Liu et al., 2017) and computer vision (Loey et al., 2021; Ganin et al., 2016; Long et al., 2017).

Despite its success in various applications, vanilla deep transfer learning approaches may suffer from the negative transfer problem (Chen et al., 2011; Ruder and Plank, 2017; Huang et al., 2007; Qu et al., 2019a; Wang et al., 2019; Chen et al., 2019), as the source and target domains usually have different data distributions. One solution to solve this problem is instance-based transfer learning, which properly selects instances from the source domain to alleviate or avoid the negative transfer. The recent instance-based transfer learning methods incorporate Reinforcement Learning (RL) for data selection (Qu et al., 2019a; Wang et al., 2019; Chen et al., 2019). These methods jointly train an RL based data selector and the transfer learning module, which is demonstrated to be better than previous methods.

However, the aforementioned methods suffer from the following two challenges. First, the transfer learning module used in the previous reinforced instance-based TL methods is a simple fully-shared model, which may not be able to learn clean domain-invariant feature representations that are discriminative in prediction (Liu et al., 2017; Shen et al., 2017). This leads to sub-optimal transfer results and the transfer learning module cannot further provide good state representations for the RL module. Second, the environment built by the transfer learning module provides sparse "delayed" rewards and with high variance during the training stage, which makes the RL policy difficult to optimize and cannot provide reasonable signals to guide the data selection process. The RL policy is performed at the instance level, but the delayed reward is at the batch level. Thus, this is a challenging sparse reward problem, i.e., to update batched sequential decisions with a single delayed reward.

In light of these challenges, we propose a **W**asserstein distance-based **S**elective **T**ransfer

* L. Feng and M. Qiu contributed equally to this work.
† Corresponding authors.

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 9772–9783
November 7–11, 2021. ©2021 Association for Computational Linguistics

Learning (WSTL) method to select helpful data from the source domain to help the target. Our method is built on top of the reinforced transfer learning framework, but differently, we introduce a Wasserstein-distance based discriminator. The advantages of the discriminator are two-fold. First, the discriminator is trained to distinguish features between the source and target, and further guides the transfer learning module to minimize the estimated Wasserstein distance in an adversarial manner. Hence, features learned in the transfer learning module are domain-invariant (i.e., more transferrable) and also discriminative in prediction (i.e., yields better transfer learning results). Second, the discriminator also provides Wasserstein distance-based metric to serve as an immediate reward signal to help guide RL policy, which can solve the sparse reward problem. In this way, the proposed method can select high-quality source data to help the target in an efficient and effective manner.

To demonstrate the benefits of the proposed WSTL method, we evaluate it on three real-world text mining tasks including paraphrase identification, natural language inference, and review text analysis. Experimental results on all these datasets show that the proposed method outperforms the state-of-the-art methods by a large margin. Empirical studies also confirm that the proposed method can select source domain data that are close to the target domain, thus indeed helps to reduce domain discrepancy and alleviate negative transfer.

We summarize the contributions as follows.

1) We proposed a Wasserstein distance based reinforced selective transfer learning method to select high-quality data efficiently and effectively to alleviate negative transfer.

2) The introduced Wasserstein discriminator provides better state representations and immediate reward signals to the reinforced selector, which leads to more stable training and better performance with faster convergence.

3) Experiments on three real-world text mining tasks demonstrate the proposed method significantly outperforms the state-of-the-art methods. The empirical studies also confirm that the data instances selected by the proposed method are closer to the target domain.

2 Proposed Method

We formulate the problem in a standard transfer learning setting. Given a source domain $D_e = \{x_i^e, y_i^e\}_{i=1}^{n_e}$ and a target domain $D_t = \{x_i^t, y_i^t\}_{i=1}^{n_t}$, our model aims to improve the performance in D_t using the rich knowledge in D_e which is usually much larger than D_t.

2.1 Model Architecture

As shown in Figure 1, our model consists of three components: a transfer learning model f_ω, a reinforced data selector f_θ and a Wasserstein distance-based discriminator f_φ. The discriminator estimates the empirical Wasserstein distance between the source domain and target domain. The TL module is then trained to minimize the estimated Wasserstein distance adversarially. The discriminator and TL module give immediate and delayed feedback respectively to guide the reinforced data selector. Details are as follows.

2.2 Wasserstein-based Discriminator

The goal for the discriminator f_φ is to distinguish source from target data. We consider using Wasserstein distance (Panaretos and Zemel, 2019) as the domain discrepancy measure, as it can provide more stable gradients even if the two distributions are distant (Arjovsky et al., 2017). Based on (Shen et al., 2017; Villani, 2008), the Wasserstein distance between source and target probability measures P_e, P_t can be approximated as:

$$W(P_e, P_t) = \frac{1}{K} \sup_{||f||_L \le K} \mathbb{E}_{x\sim P_e}\big[f(x)\big] - \mathbb{E}_{x\sim P_t}\big[f(x)\big], \tag{1}$$

where $||f||_L$ denotes Lipschitz constant. [1] Let n^e and n^t denote the number of source samples and target samples respectively, the empirical Wasserstein distance can be approximated by maximizing the discriminator loss L_{wd} when f_φ is 1-Lipschitz:

$$L_{wd}(x^e, x^t) = \frac{1}{n^e} \sum_{x^e \in X^e} f_\varphi(f_\omega(x^e)) - \frac{1}{n^t} \sum_{x^t \in X^t} f_\varphi(f_\omega(x^t)). \tag{2}$$

We enforce the Lipschitz constraint for φ by gradient penalty as suggested in Gulrajani et al. (2017). So the objective of the Wasserstein-based discriminator is as follows:

$$\max_{\varphi} \Big\{ L_{wd} + \lambda \mathbb{E}_{\hat{v}\sim P_{\hat{v}}} (||\nabla_{\hat{v}} f_\varphi(\hat{v})||_2 - 1)^2 \Big\}, \tag{3}$$

[1] For $x_1, x_2 \in X$ where X is definition domain, K is the Lipschitz constant of $f(x) \Leftrightarrow |f(x_1) - f(x_2)| \le K|x_1 - x_2|$.

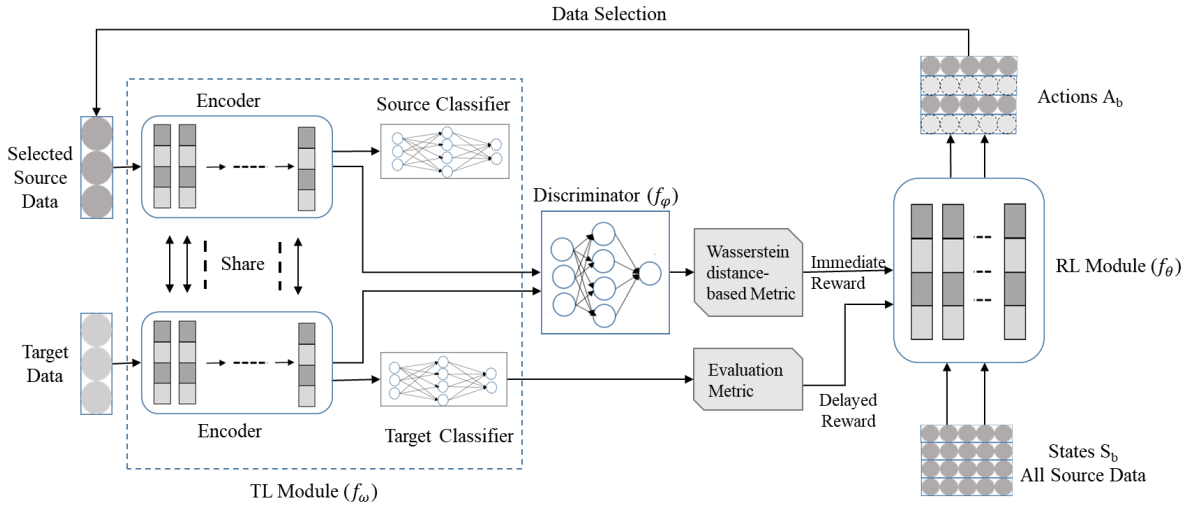


Figure 1: Overview of the proposed Wasserstein Selective Transfer Learning (WSTL) method. The Wasserstein discriminator aims to estimate the empirical Wasserstein distance between the source and target domains. The TL module is then trained adversarially with the discriminator to better learn domain invariant features. A Wasserstein distance based metric is designed to provide the immediate reward, and coupled with the delayed rewards from the evaluation environment of the TL module, to guide the RL module. Meanwhile, the RL module conducts data selection based on the state representations of the TL module.

where $P_{\hat{v}}$ is sampled uniformly along straight lines between source and target representation pairs and λ is the penalty coefficient.

2.3 TL with Wasserstein Discriminator

To make features learned in the TL module more domain invariant and suitable for transferring, the TL module and discriminator are trained adversarially: the discriminator tries to distinguish source from target data, while the TL model aims to fool the discriminator by minimizing the distances between source and target data adversarially. We adopt Wasserstein distance as the adversarial loss since the Jensen-Shannon divergence adopted in previous adversarial methods (Ganin et al., 2016; Qu et al., 2019b) suffers from discontinuities, providing less useful gradients for training. In contrast, Wasserstein distance is continuous and differentiable almost everywhere (Arjovsky et al., 2017). The superiority of Wasserstein distance for training is also verified in our experiments.

Specifically, we first train the discriminator to optimality via stochastic gradient ascent. Then we fix the optimal parameter of the discriminator and update the TL module simultaneously. Thus, the final loss of domain adversarial learning can be formulated as:

$$\min_{\omega} \Bigg\{ \sum_{k\in\{e,t\}} L_k(y, f_\omega(x)) + \beta \max_{\varphi} \Big\{ L_{wd} + \lambda \mathbb{E}_{\hat{v}\sim P_{\hat{v}}} (||\nabla_{\hat{v}} f_\varphi(\hat{v})||_2 - 1)^2 \Big\} \Bigg\}, \quad (4)$$

where $L_k(y, f_\omega(x))\,(k \in \{e,t\})$ denotes the loss of the source and target classifier. λ is set as 0 when optimizing the minimum operation since the gradient penalty should not guide the TL learning process. β is the coefficient that controls the balance between domain invariant learning and discriminative feature learning.

2.4 Reinforced Selective Training

The reinforced source data selector serves as an agent and the selection process can be modeled as a Markov decision process which can be solved by reinforcement learning: The selector selects a subset of source data, feeds into the TL module with the target data and receives rewards for this action. Selection policy π_θ is learned by interacting with the TL environment.

Specifically, let b denote batch index, n denotes the batch size, we first obtain state representation $S_b^e = \{s_1, s_2, ..., s_n\}$ for the source samples based on the semantic features generated by the shared encoder and the prediction results from the target

classifier. The action $A_b^e = \{a_1, a_2, ..., a_n\}$ where $a_i \in \{0, 1\}$ means to drop or keep the i-th instance, is decided by the policy π_θ:

$$\pi_\theta = softmax(W_2\, g(W_1 S_b^e + b_1) + b_2), \quad (5)$$

where g is ReLU activation, W_k and b_k are the weight matrix and bias of the k-th layer.

Reward Design. The objective of the data selector is to maximize the expected total reward $J(\theta)$:

$$J(\theta) = \mathbb{E}\big[R_{tot}(s, a)|\pi_\theta\big], \quad (6)$$

where θ denotes the parameters of the data selector. The RL reward signal R_{tot} consists of both immediate and delayed rewards, defined as follows.

Immediate Reward. Our immediate reward is based on the Wasserstein metric to encourage the data selector to select data instances close to the target domain. Let $L_{wd}(x_i^e, x_*^t)$ be the distance metric, it can be resolved to the following special case of the 1st Wasserstein distance:

$$\begin{aligned} L_{wd}(x_i^e, x_*^t) &= \min_{T \geq 0} \sum_{j=1}^{n} \boldsymbol{T}_{ij} d(i, j) \\ s.t. \quad & \sum_{j=1}^{n} \boldsymbol{T}_{ij} = a_i, \quad \forall i \in \{1, ..., n\}, \end{aligned} \quad (7)$$

where $d(i, j)$ denotes the Euclidean distance between x_i and x_j. Here x_i and x_j denote the output of the shared encoder in source and target domains respectively. T_{ij} measures the "travelling cost" from sample i in the source domain to sample j in the target domain. a_i denotes the binary selection action on the data instance x_i. The optimal solution is to move all probability mass of the i-th instance in X_b^e to its most similar instance j^* in X_b^t, i.e., $L_{wd}(x_i^e, x_*^t) = a_i d(i, j^*)$ where $j^* = \arg\min_j d(i, j)$. [2] To encourage the data selector to choose source data instances close to the target, the immediate reward is formulated as:

$$R_{imm} = \bar{L}_{wd}(x_*^e, x_*^t) - L_{wd}(x_i^e, x_*^t). \quad (8)$$

where $\bar{L}_{wd}(x_*^e, x_*^t)$ denotes the averaged distance between all the source and target data. The data selector seeks to find the optimal solution for $L_{wd}(x_i^e, x_*^t)$ to maximize the immediate reward. Since using the whole dataset is computationally expensive, we resolve to compute this metric at batch level to speed up the training process.

[2] Please refer to Appendix for the detailed proof.

Delayed Reward. The delayed reward is based on the evaluation metric which measures the performance difference before and after TL model updates:

$$R_b = L(y, f_\omega(x)) - L'(y, f_\omega(x)), \quad (9)$$

where $L(y, f_\omega(x))$ denotes the evaluation results of the updated model, and $L'(y, f_\omega(x))$ denotes the previous evaluation results. Based on our empirical results, we set L as the accuracy of the target for classification tasks, and as correlation coefficients between the predicted score and the ground truth score for regression tasks.

In contrast to conventional reinforcement learning, our model is updated in batches to improve the model training efficiency. For each batch in an episode, the accumulated reward is defined as:

$$R_{delay} = \sum_{k=0}^{T-t-1} \gamma^k R_{t+1+k}, \quad (10)$$

where γ is a discount factor.

Total Reward. The total reward is defined as the combination of immediate and delayed reward:

$$R_{tot} = \alpha R_{imm} + R_{delay}, \quad (11)$$

where α is the coefficient that balances the contribution of immediate reward and delayed reward.

Optimization. We adopt the policy gradient algorithm (Williams, 1992) to maximize the expected reward $J(\theta)$. The parameter θ is updated as:

$$\begin{aligned} \nabla_\theta J(\theta) &= \nabla_\theta \mathbb{E}\big[R_{tot}(s, a)|\pi_\theta\big] \\ &= \mathbb{E}_{\pi_\theta}\big[\nabla_\theta \log \pi_\theta(s, a) R_{tot}(s, a)\big] \\ &\simeq \frac{1}{K} \sum_{i=1}^{K} R_{tot}(s, a) \nabla_\theta \log \pi_\theta(s, a), \end{aligned} \quad (12)$$

where K is the selected data size.

We present the training process in Algorithm 1. Clearly, the Wasserstein based discriminator serves to (1) help the TL module to learn domain-invariant features, and provide stable state representation for RL module, and (2) provide Wasserstein distance-based metric to serve as an immediate reward signal to help guide RL policy.

Algorithm 1: Wasserstein distance-based Selective Transfer Learning (WSTL)

Require : Source and target domain data X^e, X^t, validation data V

Initialize dicriminator f_φ, TL module f_ω and data selector f_θ with random weights;
Pre-train discriminator using X^e, X^t;
Pre-train TL module using X^e, X^t;
for *each batch* (X_b^e, X_b^t) *in* (X^e, X^t) *in episode* l **do**
    Obtain representations $f_\omega(X_b^e), f_\omega(X_b^t)$
    for $i = 1, 2, \ldots, n$ **do**
        Update discriminator f_φ using Eq. (3)
    end
    Conduct data selection using Eq. (5)
    Obtain immediate reward using Eq. (8)
    Update TL module f_ω using Eq. (4)
    Obtain delayed reward using Eq. (10)
    Obtain total reward using Eq. (11)
    Update RL module f_θ using Eq. (12)
end

3 Experiments

3.1 Datasets and Implementation Details

In this paper, we conduct extensive experiments on three real-world applications, i.e., Paraphrase Identification (PI), Natural Language Inference (NLI) (Bowman et al., 2015; Qu et al., 2019a), and review helpfulness prediction (McAuley and Leskovec, 2013) to examine the effectiveness of our proposed method. For PI task, we treat the Quora question pairs [3] as the source domain and AnalytiCup [4] dataset as the target. For NLI task, we use MultiNLI (Williams et al., 2018) as the source domain and SciTail (Khot et al., 2018) as the target. Following (Chen et al., 2017), we consider Decomposable Attention Model (DAM) (Parikh et al., 2016) for PI and NLI following (Qu et al., 2019a), and TextCNN (Kim, 2014) for review helpfulness prediction . Note that the proposed method is general as we can adopt different neural architectures for TL module. The discriminator and RL module are simple neural networks with 2-hidden layers. Details about the data statics and experiment settings are in Appendix.

[3] www.kaggle.com/c/quora-question-pairs

[4] www.tianchi.aliyun.com/competition/introduction.htm?raceId=231661

Model	PI		NLI	
	ACC	AUC	ACC	AUC
Src-only	0.7538	0.5571	0.7112	0.7087
Tgt-only	0.8393	0.8548	0.7300	0.7663
TL Method	0.8488	0.8706	0.7453	0.8044
Ruder and Plank	0.8458	0.8680	0.7521	0.8062
RTL	0.8616	0.8829	0.7672	0.8163
MGTL	0.8637	0.8855	0.7782	0.8247
WSTL$_{w/o\ adv}$	0.8574	0.8856	0.7554	0.8099
WSTL$_{w/o\ RL}$	0.8691	0.9084	0.7723	0.8222
WSTL-JS	0.8631	0.8854	0.7778	0.8245
WSTL	**0.8811**†	**0.9101**	**0.7869**†	**0.8382**

Table 1: Evaluations of the proposed method on PI and NLI tasks. † denotes the statistically significant difference over the strongest baseline with $p < 0.01$ measured by the student's paired t-test. Note that AUC is an overall metric that is not suitable for t-test.

3.2 Experiments on PI and NLI

We compare our model with these baseline models.

- **Src-only**: A model only trained on source domain and tested on the target domain.
- **Tgt-only** (Parikh et al., 2016): A model only uses target domain data described in Sec. 2.4.
- **TL Method**: a typical TL method in (Mou et al., 2015, 2016) that uses a shared encoder and domain-specific output layers.
- **Ruder and Plank** (Ruder and Plank, 2017): an instance selection method with Bayesian optimization.
- **RTL**(Qu et al., 2019a): a recent proposed RL based instance selection method.
- **MGTL**(Wang et al., 2019): another recent generative adversarial network based instance selection method.
- **WSTL**$_{w/o\ adv}$: degenerated versions of our model which does not include the adversarial training process between the discriminator and TL module.
- **WSTL**$_{w/o\ RL}$: degenerated versions of our model, without the reinforced data selector.
- **WSTL-JS**: a variant of our method with JS divergence as distance metrics.

As in Table 1, we have several observations.

(1) We observe that the proposed WSTL method achieves the best performance and significantly outperforms other methods on both tasks, which shows the superiority and effectiveness of the method.

(2) Due to the domain shift, Src-only performs worse than Tgt-only. The TL method outperforms the Tgt-only model, for leveraging information from the source domain can help the target domain.

(3) Existing instance selection methods such as RTL and MGTL have improved performance over TL, which shows that data selection can alleviate negative transfer. WSTL outperforms all competing methods, indicating the proposed Wasserstein-based discriminator can help both RL and TL to stabilize training and promote better selection.

(4) The improvement over two variants of our model WSTL$_{w/o\ adv}$ and WSTL$_{w/o\ RL}$ also shows the importance of Wasserstein-based adversarial training and RL module.

(5) Compared with JS divergence, the proposed WSTL method outperforms WSTL-JS, which shows the effectiveness of the Wasserstein distance metric.

In general, by integrating the TL module, RL module and the Wasserstein-based discriminator in a unified framework, WSTL can significantly outperform the competing methods on all tasks.

Correlation	Watch	Phone	Outdoor	Home
STR	0.276	0.349	0.277	0.222
UGR	0.425	0.466	0.412	0.309
LIWC	0.378	0.464	0.382	0.331
INQ	0.403	0.506	0.419	0.366
ASP	0.406	0.437	0.385	0.283
Fusion 1	0.488	0.539	0.497	0.432
Fusion 2	0.493	0.550	0.501	0.436
Src-only	0.483	0.443	0.449	0.395
Tgt-only	0.559	0.556	0.640	0.564
TL Method	0.557	0.560	0.651	0.570
TL-adv	0.501	0.564	0.511	0.468
TL-dd	0.515	0.571	0.510	0.472
RTL	0.564	0.566	0.654	0.573
MGTL	0.571	0.567	0.656	0.574
WSTL$_{w/o\ adv}$	0.572	0.565	0.653	0.571
WSTL$_{w/o\ RL}$	0.574	0.563	0.652	0.572
WSTL-JS	0.570	0.567	0.655	0.574
WSTL	**0.598**	**0.575**	**0.660**	**0.579**

Table 2: Results for review helpfulness prediction.

3.3 Experiments on Review Helpfulness Prediction

Review helpfulness prediction is a regression task that predicts the helpfulness score of a given review. We compare our model with regression baselines that use hand-crafted features which are STR, UGR, LIWC, INQUIRER, aspect-based feature ASP, and two groups of ensemble features named Fusion1 and Fusion2 from Chen et al. (2018a). We also compare with Src-only, Tgt-only (Kim, 2014), Vanilla TL Method, and two degenerated versions of our model WSTL$_{w/o\ adv}$ and WSTL$_{w/o\ RL}$ as described in Section 3.2. In addition, we compare with two recent proposed TL methods for the task:

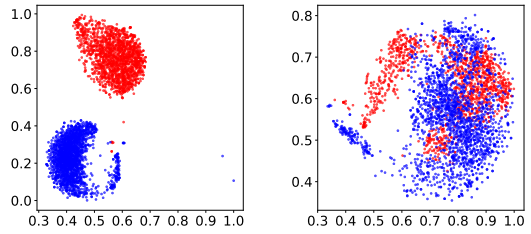
- **TL-dd** (Chen et al., 2018a): a cross-domain model with auxiliary domain discriminators.
- **TL-adv** (Liu et al., 2017): TL method with adversarial training to alleviate the shared and private features from interfering each other.

As shown in Table 2, we have similar observations with the experiments on PI and NLI. (1) Those traditional methods without using deep neural networks, e.g., from STR, UGR to Fusion methods, cannot perform as well as deep learning methods. This shows deep learning methods have the ability to extract more important semantic features for the task. (2) Source and target domain data have similar but different data distributions since Src-only performs worse than Tgt-only. (3) Transfer learning methods significantly outperform "non-transfer" methods and hand-crafted features, which demonstrate that the performance of the target domain can be improved with a large margin with the help of transfer learning. (4) WSTL outperforms two state-of-the-art TL methods TL-dd and TL-adv, indicating the importance of source data selection on the task. (5) WSTL achieves 1% improvement on average over RTL and MGTL methods, demonstrating that domain invariant features and Wasserstein distance-based metric can circumvent the difficulty of RL training. (6) We find both adversarial training with discriminator and RL module are important for the task, as our model outperforms two degenerated models. (7) Our method with Wasserstein distance achieves improvement over WSTL-JS on different domains, which shows the superiority of Wasserstein distance over other distance metrics.

3.4 Effects of the Wasserstein Discriminator

To demonstrate the effectiveness of the Wasserstein discriminator, we compare our method with the vanilla RTL method (Qu et al., 2019a).

Distances between Domains. We first compare the distances between the selected data and the target data on "Watch" task. The Wasserstein distances for the original data, selected data from RTL, and selected data from our WSTL are 3.841e-06, 3.782e-06, and 3.436e-06, respectively. Clearly, both WSTL and RTL methods can reduce distances between source and target data distribution, while WSTL can select data instances closer to the target.



(a) RTL (b) WSTL

Figure 2: Feature visualization of different methods. Red and blue points denote source and target data.

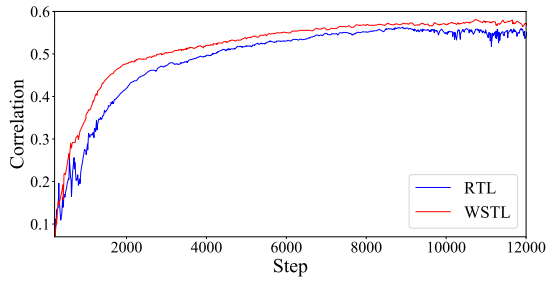


Figure 3: Convergence curve for RTL and WSTL.

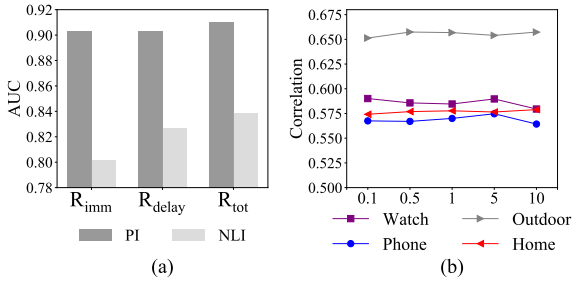


Figure 4: Model performance w.r.t. (a) different rewards on both PI ad NLI tasks and (b) different values of hyper-parameter α.

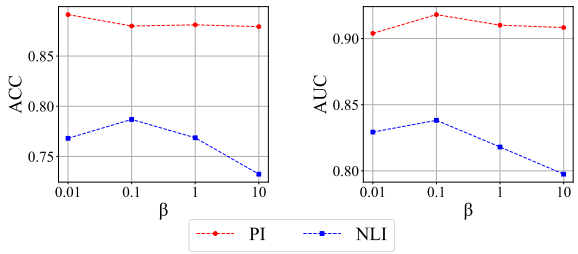


Figure 5: Model performance w.r.t. different values of hyper-parameter β

Feature Visualization. We use t-SNE to demonstrate the proposed method's capacity of minimizing the divergence between source and target distributions. As shown in Figure 2, we observe that instances from different domains are closer in Figure 2(b). It corroborates that by adopting Wasserstein discriminator, WSTL can help to learn domain invariant features which can effectively reduce domain discrepancy. The detailed feature distributions at different training steps are in Appendix. Moreover, WSTL improves around 2% over RTL in Table 1 and 1% on average in Table 2, which shows WSTL can learn discriminative features with better results on target domain.

Convergence and Stability. We compare the convergence and stability between WSTL and RTL methods on "Watch" task in review helpfulness prediction. As shown in Figure 3, we can observe that WSTL converges faster and performs more stably than the RTL method. The reason is that Wasserstein distance-based discriminator can stabilize learning and provide domain invariant representation and Wasserstein distance-based metric to help reinforced data selector learn more efficiently. We also find WSTL performs better than RTL method since better state representations and reward signals provided by Wasserstein distance-based domain discriminator can make WSTL select high-quality source data to improve performance in the target domain.

3.5 Parameter Sensitivity Analysis

Effect of different rewards. We test the effect of different rewards on the data selector learning.

As in Figure 4(a), we can observe that both delayed reward and immediate reward are helpful for the task, and the best performance can be achieved by combining with both types of rewards.

We also demonstrate the impact of hyper-parameter α which measures the contribution of immediate reward and delayed reward. As in Figure 4(b), we observe that our method is generally robust to different values of the hyper-parameter α as it shows to have similar performance with different settings.

Effect of Feature Learning. We tested the impact of hyper-parameter β which controls the balance between domain invariant learning and discriminative feature learning. We tested with different values of hyperparameter $\beta = \{0.01, 0.1, 1, 10\}$ in Equation 4. As shown in Figure 5, we can observe that the performance on PI task is less sensitive to parameter β than NLI task.

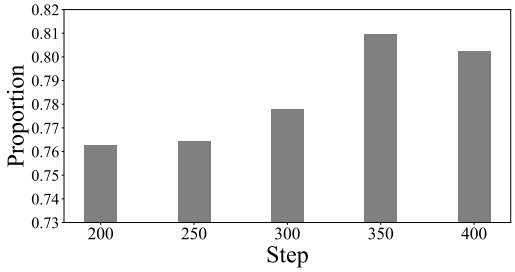


Figure 6: The proportion of "address" in selected data with the same semantic meaning as in target domain.

3.6 Case Study

To examine whether our model can select desirable samples from the source domain, we design an experiment to examine the behaviors of the data selector. In source NLI data, the term "address" has two meanings, i.e., a verb that begins to deal with something, and a noun that describes a location. However, in target NLI data, the term is mostly about the first meaning. Hence we remove sentences with the second meaning in the target and collect examples containing "address" in both datasets, to examine whether the proposed method can select proper sentences. We compute the proportion of selected source examples containing the same semantic meaning of "address" as in SciTail.

From Figure 6, we observe that with model training, the data selector can select more instances with the same "address" meaning as in SciTail. This result is insightful as it shows the proposed method can transfer relevant examples to promote positive transfer and ignore irrelevant ones to mitigate negative transfer.

4 Related Work

Transfer learning (TL). TL has been widely studied in various applications (McCann et al., 2017; Deng et al., 2009; Yosinski et al., 2014). Due to the domain difference, a vanilla transfer learning method may suffer from negative transfer. There are generally two lines of study to address this problem. The first is feature-based methods, which aim to locate a common feature space that can reduce the differences between the source and target domains (Shen et al., 2017). However, the capacity of shared space could be consumed by some unnecessary features. The second category is instance-based methods, which re-weight the source samples so that data from the source and target domain share a similar data distribution (Chen et al., 2011; Huang et al., 2007; Ruder and Plank, 2017). The TL module is typically considered as a sub-module of the data selection framework (Ruder and Plank, 2017). Therefore, the TL module needs to be retrained repetitively to provide sufficient updates to the selection framework which may suffer from long training time. The recent studies (Qu et al., 2019a; Wang et al., 2019) consider RL based instance selection methods to jointly train the data selector and the TL model. However, the training of the TL module struggles to maintain stable and the mislead signal from the TL model inevitably increases the difficulty of data selection. Our method employs the Wasserstein discriminator to help both TL and RL modules. It provides domain invariant features to serve as states for the RL policy and helps to improve TL module via adversarial training. The Wasserstein discriminator also provides immediate rewards to guide the RL policy.

Different from several domain adaption methods proposed for sentiment classification (Qu et al., 2019b; Du et al., 2020; Xue et al., 2020; Zhang et al., 2019; He et al., 2018) where the labels in the target domain are not available, both the source and target labels are available in our transfer learning setting. In our setting, we seek to leverage data-sufficient domains to help target domains with less sufficient data labels. Besides, these methods are designed for sentiment classification while our method is more general on various task such as PI and NLI.

Wasserstein Distance. The Wasserstein distance (Panaretos and Zemel, 2019) is a metric based on the theory of optimal transport. It gives a natural measure of the distance between two probability distributions. Arjovsky et al. (2017) introduce Wasserstein metric to alleviate the vanishing gradient and the mode collapse issues in the original GAN (Goodfellow et al., 2014). Chen et al. (2018b) propose to minimize the Wasserstein distance between different domains for cross-lingual sentiment classification. Shen et al. (2017) adopts Wasserstein distance to representation learning for domain adaptation. However, the learned representations are contaminated by misleading features, suffering from feature redundancy. Yu et al. (2020) introduces Wasserstein distance as a regularizer to improve the sequence representations. Inspired by its success in various applications, we introduce Wasserstein distance to selective transfer learning.

5 Conclusions

To alleviate the negative transfer issues, we propose a Wasserstein Selective Transfer Learning (WSTL) method that builds a Wasserstein discriminator to maximize the empirical distance between the selected source and target domain data. The TL module is trained to minimize the estimated Wasserstein distance in an adversarial manner, and the discriminator provides immediate rewards further coupled with the delayed rewards from the TL module to guide the reinforced data selector. Extensive experiments on three real-world datasets show the proposed method significantly outperforms the competing methods.

Acknowledgments

This research is supported by National Natural Science Foundation of China (Grant No. 61773229 and 6201101015), Alibaba Innovation Research (AIR) programme, Natural Science Foundation of Guangdong Province (Grant No. 2021A1515012640), the Basic Research Fund of Shenzhen City (Grand No. JCYJ20210324120012033 and JCYJ20190813165003837), and Overseas Cooperation Research Fund of Tsinghua Shenzhen International Graduate School (Grant No. HW2021008).

References

Martin Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein gan. *arXiv preprint arXiv:1701.07875*.

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. *EMNLP*.

Cen Chen, Yinfei Yang, Jun Zhou, Xiaolong Li, and Forrest Bao. 2018a. Cross-domain review helpfulness prediction based on convolutional neural networks with auxiliary domain discriminators. In *NAACL*, pages 602–607.

Minmin Chen, Kilian Q Weinberger, and John Blitzer. 2011. Co-training for domain adaptation. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2456–2464.

Xilun Chen, Yu Sun, Ben Athiwaratkun, Claire Cardie, and Kilian Weinberger. 2018b. Adversarial deep averaging networks for cross-lingual sentiment classification. *Transactions of the Association for Computational Linguistics*, 6:557–570.

Xinchi Chen, Zhan Shi, Xipeng Qiu, and Xuanjing Huang. 2017. Adversarial multi-criteria learning for chinese word segmentation. *ACL*.

Zhihong Chen, Chao Chen, Zhaowei Cheng, Ke Fang, and Xinyu Jin. 2019. Selective transfer with reinforced transfer network for partial domain adaptation. *CoRR*, abs/1905.10756.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition (CVPR)*, pages 248–255. Ieee.

Chunning Du, Haifeng Sun, Jingyu Wang, Qi Qi, and Jianxin Liao. 2020. Adversarial and domain-aware bert for cross-domain sentiment analysis. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4019–4028.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030.

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 (NeurIPS)*, pages 2672–2680.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. 2017. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 5767–5777.

Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. 2018. Adaptive semi-supervised learning for cross-domain sentiment classification. *arXiv preprint arXiv:1809.00530*.

Jiayuan Huang, Arthur Gretton, Karsten Borgwardt, Bernhard Schölkopf, and Alex J Smola. 2007. Correcting sample selection bias by unlabeled data. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 601–608.

Tushar Khot, Ashish Sabharwal, and Peter Clark. 2018. Scitail: A textual entailment dataset from science question answering. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *EMNLP*.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *International Conference on Learning Representations*.

Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2017. Adversarial multi-task learning for text classification. *ACL*.

Mohamed Loey, Gunasekaran Manogaran, Mohamed Hamed N Taha, and Nour Eldeen M Khalifa. 2021. A hybrid deep transfer learning model with machine learning methods for face mask detection in the era of the covid-19 pandemic. *Measurement*, 167:108288.

Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. 2017. Deep transfer learning with joint adaptation networks. In *ICML*, pages 2208–217. JMLR. org.

Lionel Martin and Pearl Pu. 2014. Prediction of helpful reviews using emotions extraction. In *AAAI*, pages 1551–1557.

Julian McAuley and Jure Leskovec. 2013. Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proceedings of the 7th ACM conference on Recommender systems*, pages 165–172.

Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2017. Learned in translation: Contextualized word vectors. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 6294–6305.

Lili Mou, Rui Men, Ge Li, Yan Xu, Lu Zhang, Rui Yan, and Zhi Jin. 2015. Natural language inference by tree-based convolution and heuristic matching. *ACL*.

Lili Mou, Zhao Meng, Rui Yan, Ge Li, Yan Xu, Lu Zhang, and Zhi Jin. 2016. How transferable are neural networks in nlp applications? *EMNLP*.

Sinno Jialin Pan and Qiang Yang. 2009. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359.

Victor M Panaretos and Yoav Zemel. 2019. Statistical aspects of wasserstein distances. *Annual review of statistics and its application*, 6:405–431.

Ankur P Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. 2016. A decomposable attention model for natural language inference. *EMNLP*.

Minlong Peng, Qi Zhang, Yu-gang Jiang, and Xuan-Jing Huang. 2018. Cross-domain sentiment classification with target domain specific information. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2505–2513.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Chen Qu, Feng Ji, Minghui Qiu, Liu Yang, Zhiyu Min, Haiqing Chen, Jun Huang, and W Bruce Croft. 2019a. Learning to selectively transfer: Reinforced transfer learning for deep text matching. In *WSDM*, pages 699–707.

Xiaoye Qu, Zhikang Zou, Yu Cheng, Yang Yang, and Pan Zhou. 2019b. Adversarial category alignment network for cross-domain sentiment classification. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2496–2508.

Sebastian Ruder and Barbara Plank. 2017. Learning to select data for transfer learning with bayesian optimization. *EMNLP*.

Jian Shen, Yanru Qu, Weinan Zhang, and Yong Yu. 2017. Wasserstein distance guided representation learning for domain adaptation. *AAAI*.

Cédric Villani. 2008. *Optimal transport: old and new*, volume 338. Springer Science & Business Media.

Bo Wang, Minghui Qiu, Xisen Wang, Yaliang Li, Yu Gong, Xiaoyi Zeng, Jun Huang, Bo Zheng, Deng Cai, and Jingren Zhou. 2019. A minimax game for instance based selective transfer learning. In *SIGKDD*.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *NAACL*, pages 1112–1122.

Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256.

Qianming Xue, Wei Zhang, and Hongyuan Zha. 2020. Improving domain-adapted sentiment classification by deep adversarial mutual learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9362–9369.

Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. 2014. How transferable are features in deep neural networks? In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 3320–3328.

Weijie Yu, Chen Xu, Jun Xu, Liang Pang, Xiaopeng Gao, Xiaozhao Wang, and Ji-Rong Wen. 2020. Wasserstein distance regularized sequence representation for text matching in asymmetrical domains. *arXiv preprint arXiv:2010.07717*.

Kai Zhang, Hefu Zhang, Qi Liu, Hongke Zhao, Hengshu Zhu, and Enhong Chen. 2019. Interactive attention transfer network for cross-domain sentiment classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 5773–5780.

A Appendices.

A.1 Proof for Equation 7

The optimal solution is to move all probability mass of each instance in X_b^e to its most similar instance in X_b^t. Precisely, an optimal $\boldsymbol{T}^*$ matrix is defined as:

$$T_{ij}^* = \begin{cases} a_i & \text{if } j = \arg\min_j d_{i,j}, \\ 0 & \text{otherwise.} \end{cases} \quad (13)$$

Below we derive the solution.

PROOF. Let $j^* = \arg\min_j d(i,j)$, then we have:

$$\begin{aligned} \sum_j T_{ij} d(i,j) &\geq \sum_j T_{ij} d(i,j^*) \\ &= d(i,j^*) \sum_j T_{ij} \\ &= d(i,j^*) a_i \\ &= \sum_j T_{ij}^* d(i,j). \end{aligned}$$ ■

Therefore, T^* must yield a minimum objective value. Thus, for each instance vector x_i in X_b^e, we can achieve the minimum objective value by a nearest neighbor search to obtain j^* where $j^* = \arg\min_j d(i,j)$, and obviously the distance metric $L_{wd}(x_i^e, x_*^t) = a_i d(i,j^*)$.

A.2 Experiment Settings

PI task. This is a typical text matching task that is widely used in dialogue systems (Qu et al., 2019a). The task is to examine the relationship, i.e., paraphrase or not, between two input text sequences. We treat the Quora question pairs [5] as the source domain and AnalytiCup [6] dataset as the target. The former is a large-scale dataset that covers a variety of topics which has 404287 examples, while the latter consists of question pairs from E-commerce which has 6668 examples in training set, 3334 examples in validation set and 3330 examples in test set. We follow the study in (Qu et al., 2019a) for data preprocessing.

NLI task. This is a natural language inference task to examine whether the semantic relation indicates whether a hypothesis can be inferred from a premise (Bowman et al., 2015). We use MultiNLI as the source domain and SciTail as the target. The former is a large crowd-sourced benchmark corpus from a wider range of text genres which has 261799 examples. The latter is a recently released challenging textual entailment dataset collected from the science domain which has 23596 examples in training set, 1304 examples in validation set and 2126 examples in test set.

Dataset	Domain	# of reviews (> 5 votes)
Electronics	source	354,301
Watches	target	9,737
Cellphones	target	18,542
Outdoor	target	72,796
Home	target	219,310

Table 3: Amazon reviews from 5 different domain categories.

Review helpfulness prediction. This is a text mining task to examine the helpfulness score of a given review. Due to the high volume of reviews on E-commerce sites, it's an important task that draws increasing attention from both academia and industry (Martin and Pu, 2014). We use reviews from five categories of products in the Amazon review dataset (McAuley and Leskovec, 2013). The data from the Electronics domain (the largest dataset) are served as the source domain, while the rest four domains are treated as target domains. Data statistics are summarized in Table 3.

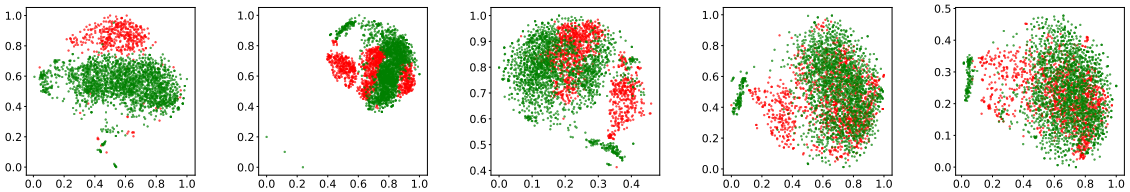
The experiments are conducted on a Linux server equipped with an Intel(R) Xeon(R) CPU E5-2699 v4 @ 2.20GHz and 8 NVIDIA V100-SXM2-16GB GPUs. We implement our model via TensorFlow and the models are trained with Adam optimizer (Kingma and Ba, 2014). The discriminator is designed with a hidden layer of 128 nodes. The learning rate for the discriminator is $1e{-}4$ and the training steps n is set to 5 for fast computation and sufficient optimization guarantee for the discriminator. The penalty coefficient λ is set to 10 as suggested in (Gulrajani et al., 2017).

For both PI and NLI tasks, the size for the hidden layers of the decomposable attention model is 256. The max sequence length is set to 50. Word embeddings are initialized with GloVe word vectors (Pennington et al., 2014) and are set to trainable. The initial learning rate is set as 0.001. The transfer learning model is pre-trained for 50 iterations before the reinforced data selector is applied.

For review helpfulness prediction task, the lookup table is also initialized with pre-trained vec-

[5] www.kaggle.com/c/quora-question-pairs

[6] www.tianchi.aliyun.com/competition/introduction.htm?raceId=231661



(a) Step: $4e3$ (b) Step: $6e3$ (c) Step: $8e3$ (d) Step: $10e3$ (e) Step: $12e3$

Figure 7: Feature visualization of WSTL at different training steps.

tors from Glove. For CNN, the activation function is ReLU, and the channel size is set to 128. Multiple filters are used here with window size $l \in \{2, 3, 4, 5\}$. We use Adam as the optimizer. Following the previous work (Chen et al., 2018a), ten-fold cross-validation is performed for all experiments and all the results are evaluated in correlation coefficients between the predicted helpfulness score and the ground truth score computed by "a of b approach" from the dataset.

A.3 Feature Visualization

To demonstrate the transferability of the selected features, we use t-SNE to visualize the selected feature representation from the source domain and target domain. We choose the "Watch" task in review helpfulness prediction as an example. In Figure 7, red points denote samples from the source domain, green points denote samples from the target domain. We can observe that with the model training, red and green points are getting closer, which denotes that our model can select useful source data instances that are close to the target domain. It proves that adopting Wasserstein distance in adversarial training can help learn domain invariant features, and such features can help to effectively alleviate negative transfer.