TSAL: Two steps Adversarial learning based domain adaptation

Haidi Badr [1], **Nayer wanas**[1], **Magda Fayek**[2]
[1] Electronics Research Institute, Informatics Dapartment, Egypt
[2]Cairo university, Faculty of Engineering, Computer Department, Egypt

Abstract

Since labeled data availability differs greatly across domains, Domain Adaptation focuses on learning in new and unfamiliar domains by reducing distribution divergence. Recent research suggests that the adversarial learning approach could be a promising way to achieve the domain adaptation objective. Adversarial learning is a strategy for learning domain-transferable features in robust deep networks. This paper introduces the TSAL paradigm, a two-step adversarial learning framework. It addresses the real-world problem of text classification, where source domain(s) has labeled data but target domain (s) has only unlabeled data. TSAL utilizes joint adversarial learning with class information and domain alignment deep network architecture to learn both domain-invariant and domain-specific features extractors. It consists of two training steps that are similar to the paradigm, in which pre-trained model weights are used as initialization for training with new data. TSAL's two training phases, however, are based on the same data, not different data, as is the case with fine-tuning. Furthermore, TSAL only uses the learned domain-invariant feature extractor from the first training as an initialization for its peer in subsequent training. By doubling the training, TSAL can emphasis the leverage of the small unlabeled target domain and learn effectively what to share between various domains. A detailed analysis of many benchmark datasets reveals that our model consistently outperforms the prior art across a wide range of dataset distributions.

Introduction

Domain Adaptation (DA) is an area that deals with the problem of training a model with data from one or more source domains, then transferring the information to a new target domain with different data distribution. The new target domain may contain entirely unlabeled samples (unsupervised domain adaptation), fully labeled samples (supervised domain adaptation), or a small number of labeled samples (semi-supervised domain adaptation). DA intends to minimize the distance between the source and target domains as much as possible, making it easier to transfer between them. It has been used for sentiment classification (Zhang, Xu, and Hu 2015; Zhou et al. 2016a), Part−of−Speech (POS) tagging (Schnabel and Schutze 2013), information retrieval (Zhou et al. 2016b; 2018), and machine translation (Gu et al. 2019), as well as several other NLP tasks. For instance, in the context of sentiment analysis in written reviews, one may have labeled data for one type of product (e.g., movies) while needing to identify reviews for other products (e.g., dvd). DA approaches have recently been used to narrow the gap between the source and target feature distributions (Yosinski et al. 2014; Chen et al. 2012; Vincent et al. 2008; Tzeng et al. 2017; Liu, Qiu, and Huang 2017; Ganin et al. ; Zhao et al. 2017). Adversarial learning is a state-of-the-art framework for deep architectures-based DA (Tzeng et al. 2017; Zhao et al. 2017). In particular, the Multinomial Adversarial Networks (MANs) that proposed by Chen and Cardie (Chen and Cardie 2018). It adopts the shared-private model (Bousmalis et al. 2016) to learn domain-invariant and domain-specific features, using a shared and domain feature extractor.

As compared to other DA methods, the MAN model achieves state-of-the-art performance when it comes to the unsupervised domain adaptation, (Chen and Cardie 2018). In this paper, we suggest TSAL, two steps domain adversarial learning, which adopts the MAN model through a couple of learning steps. The fine-tuning model is at the heart of the TSAL premise. The TSAL model's first step is identical to that of the MAN model, which starts with random weights and eventually trains to a good domain-invariant representation. However, due to the scarcity of target samples, the amount of information that can be learned about the target is limited, particularly when the distance between the source and target distribution is significant. As a consequence, we suggest that using the first step's learned features as initialization in the second step training will assist in revealing more insightful target features. On various benchmark datasets, we validate the efficacy of TSAL. It strengthens the MAN and other state-of-the-art methods thus achieving a substantial increase in average accuracy.

Related work

Depending on the availability of labeled data and the cost of collecting it, DA methods can be categorized into supervised DA, semi-supervised DA (Mathapati et al. 2019), and unsupervised DA (Wilson and Cook 2019).

Daume (Daumé III 2007) defines supervised DA as the case where both the source and target domains have an abundance of labeled data. Conversely, semi-supervised DA is when there is sufficiently labeled data from the source domain but only minimal labeled data from the target domain (Xiao and Guo 2015; Blitzer, McDonald, and Pereira 2006). Finally, unsupervised DA (UDA) is used when there is a lot of labeled data from the source domain but only unlabeled data from the target domain(Kouw and Loog 2019; Das and Lee 2018; Gong et al. 2012), which is the scope of this paper. The key strategy of UDA is to keep the gap between the source and target as small as possible.

The literature has mostly focused on a linear hypothesis over the years (Blitzer, McDonald, and Pereira 2006; Baktashmotlagh et al. 2013a; 2013b). Joaqui et.al., for example, (Quiñonero-Candela et al. 2008) used the Maximum Mean Discrepancy (MMD) loss to minimize domains discrepency. Tzeng et.al., (Tzeng et al. 2014) used MMD with a regular classification loss on the source to learn a representation that is both discriminative and domain invariant. MMD and reproducing kernel Hilbert space were also used by Long et.al., (Long et al. 2015) also Long et.al., (Long et al. 2015) to align higher order statistics of the two distributions effectively.

More recently, UDA approaches are largely built on modern deep architectures to minimize the difference between the source and target feature distributions (Yosinski et al. 2014; Chen et al. 2012; Vincent et al. 2008). One of the earliest studies using deep learning for domain adaptation was Stacked Denoising Autoencoders (SDA), proposed by Vincent et.al., (Chen et al. 2012). It's an auto-encoder-based method that trains the model to minimize the loss between the original inputs and their reconstructions.

Another line of research is domain-adversarial training, which is the current state of the art (Wang et al. 2018; Ganin et al. 2016; ; Chen and Cardie 2018; Li et al. 2017). To minimize domain shift, these approaches used an adversarial loss, learning a representation that is both discriminative of source labels and unable to differentiate between domains. An external domain classifier is augmented with a classification network. Adversarial training aims to learn domain-invariant features, by playing a minimax game (Goodfellow et al. 2014), (Liu, Qiu, and Huang 2017; Tzeng et al. 2017).

Chen and Cardie (Chen and Cardie 2018) propose a multinomial adversarial network (MAN) that learns invariant features across multiple domains. It adopts the Shared-Private paradigm of Bousmalis et al. (Bousmalis et al. 2016) to learn domain-invariant features. MAN is the motivation for our proposed TSAL. TSAL can be considered as a two-step of MAN. It has the potential to be more reliable than MAN and related techniques.

Model

Architecture

In this paper, we aim to tackle the DA in the context of a text classification problem. For source domains, we have a mix of labeled and unlabeled data, while for the target domain, we only have unlabeled data. Using an adversarial learning approach, the TSAL is proposed to ease the knowledge transfer from a seen domain (source) to an unseen domain (target). As shown in Figure 1, TSAL architecture consists of two identical subsequence steps. Each step follows the same MAN architecture (Chen and Cardie 2018), that uses the Shared-Private paradigm.

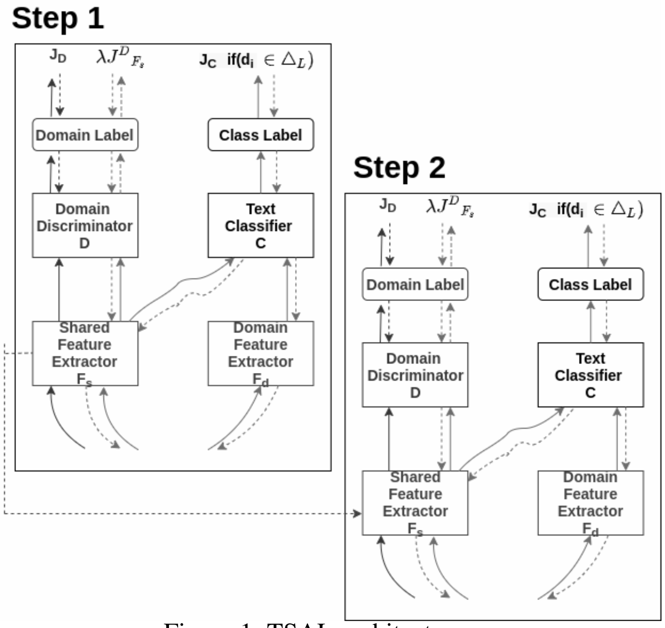

Figure 1: TSAL architecture

Step 1 consists of four components: a shared feature extractor F$_s$, a domain feature extractor F$_{d_i}$ for each source domain, a text classifier C, and a domain discriminator D. The intuitive is that domain-invariant features that are beneficial to the main classification task in all domains (i.e. the shared features, extracted by F_s), as well as the domain-specific features that mainly contribute to the classification in its domain (the domain features, extracted by F$_d$), are explicitly modeled.

The adversarial domain discriminator D is a domain classifier that predicts which domain an input sample belongs to based on the extracted shared features. Both F_s and D play the mini-max game, in which F_s attempts to confuse D by minimizing its loss, so that D is unable to predict the domain of a sample given its shared features. The hypothesis is that if D can't recognize the input's domain, the shared features don't contain domain knowledge and are therefore domain-invariant. It worth noting that D is only trained with unlabeled samples, therefore it's trained on both the source and target domains. By forcing domain-invariant features to be learned by F_s, the set of domain feature extractors F_d will each learn domain-specific features beneficial within its domain when trained jointly through back-propagation. The text classifier C module is responsible for making predictions for the end task. For each input, it first concatenates the shared and private features into a single feature vector. The input's sample class label is then predicted. Each component's architecture can be chosen in a variety of ways. Depending on the input data, feature extractors may be Convolutional Neural Nets (CNN), Recurrent Neural Nets (RNN),

	MLP	mSDA	DANN	MDAN (H-MAX)	MDAN (S-MAX)	MAN	DACL	TSAL
Books	76.55	76.98	77.89	78.45	78.63	77.78	**80.22**	**80.2**
DVD	75.88	78.61	78.86	77.97	80.65	82.74	**82.96**	82.7
Elec.	84.60	81.98	84.91	84.83	85.34	83.75	84.9	**85.3**
Kit	85.45	84.26	86.39	85.80	86.26	86.42	86.75	**87.0**
AVG.	80.46	80.46	82.01	81.76	82.72	82.30	83.71	**83.8**

Table 1: Unsupervised domain adaptation results on the Amazon review dataset. Highest domain performance is shown in bold.Except for our model, the rest is taken from (Chen and Cardie 2018).

	books	Elec.	dvd	Kitch.	apparel	camera	health	music	toys	video	baby	magaz.	softw.	sports	IMDb	MR	AVG.
MAN	86.62	88.25	86.25	89.25	85.25	86.25	83.75	86.25	86.0	83.3	79.75	85.25	80.5	80.5	80.75	71.5	83.71
TSAL	87.25	89.25	87.25	89.5	86.0	87.0	86.25	86.5	88.75	85.5	84.75	86.5	83.5	85.75	86.75	74.5	85.94

Table 2: Unsupervised domain adaptation results on the FDU-MTL review dataset. Multi-source domains V.S. one target domain.

or a Multi-Layer Perceptron (MLP). A fixed-length vector representing the (shared/domain) hidden features of a given input sample is the output of a (shared/domain) feature extractor. On the other hand, the outputs of C and D are class and domain label probabilities, respectively.

Step 2 is the key distinction between our proposed model and MAN (Chen and Cardie 2018). It contains the same components as step 1, however in this step, the learned shared extractor (F_s) from step 1 is used as the initialization for learning its peer.

The TSAL model hypothesis is based on the fine-tuning paradigm. As fine-tuning deep learning involves using the weights of a previous deep learning algorithm as initialization for training a new deep learning algorithm. TSAL uses weights from a previous deep learning algorithm, F_s of step 1, for initialization of another related deep learning algorithm, F_s of step 2. step 2 is guided to start learning from the previously learned shared representation. We expect that step 2 will reveal more deep features that support the target domain. The results of the experiments support this hypothesis. When compared to MAN and other related approaches, TSAL has a positive impact on performance.

Training

The MAN training procedure (Chen and Cardie 2018) is described in Algorithm 1. The training flows of different components are depicted by the arrows. The domain discriminator D is trained with a separate optimizer due to its adversarial nature, while the rest of the networks are updated with the main optimizer.

Only annotated data from source domains are used to train the text classifier C, which takes the concatenation of shared and domain feature vectors as input. The domain features for C's input is set to a 0 vector at test time for data from unlabeled domains with no F_d. For both labeled and unlabeled domains, D, on the other hand, only takes the shared features as input. For the loss functions of C (L_C) and D(L_D) the canonical negative log-likelihood loss is used.

Algorithm 1 MAN Training

```
Require: labeled corpus $\mathbb{X}$; unlabeled corpus $\mathbb{U}$; Hyperpamameter $\lambda > 0$, $k \in \mathbb{N}$
repeat
    ▷ $\mathcal{D}$ iterations
    for $diter = 1$ to $k$ do
        $l_{\mathcal{D}} = 0$
        for all $d \in \Delta$ do                          ▷ For all $N$ domains
            Sample a mini-batch $\boldsymbol{x} \sim \mathbb{U}_d$
            $\boldsymbol{f}_s = \mathcal{F}_s(\boldsymbol{x})$                       ▷ Shared feature vector
            $l_{\mathcal{D}} \mathrel{+}= J_{\mathcal{D}}(\mathcal{D}(\boldsymbol{f}_s); d)$      ▷ Accumulate $\mathcal{D}$ loss
        Update $\mathcal{D}$ parameters using $\nabla l_{\mathcal{D}}$
    ▷ Main iteration
    $loss = 0$
    for all $d \in \Delta_L$ do                   ▷ For all labeled domains
        Sample a mini-batch $(\boldsymbol{x}, \boldsymbol{y}) \sim \mathbb{X}_d$
        $\boldsymbol{f}_s = \mathcal{F}_s(\boldsymbol{x})$
        $\boldsymbol{f}_d = \mathcal{F}_d(\boldsymbol{x})$                       ▷ Domain feature vector
        $loss \mathrel{+}= J_{\mathcal{C}}(\mathcal{C}(\boldsymbol{f}_s, \boldsymbol{f}_d); \boldsymbol{y})$       ▷ Compute $\mathcal{C}$ loss
    for all $d \in \Delta$ do                          ▷ For all $N$ domains
        Sample a mini-batch $\boldsymbol{x} \sim \mathbb{U}_d$
        $\boldsymbol{f}_s = \mathcal{F}_s(\boldsymbol{x})$
        $loss \mathrel{+}= \lambda \cdot J_{\mathcal{F}_s}^{\mathcal{D}}(\mathcal{D}(\boldsymbol{f}_s); d)$   ▷ Domain loss of $\mathcal{F}_s$
    Update $\mathcal{F}_s, \mathcal{F}_d, \mathcal{C}$ parameters using $\nabla loss$
until convergence
```

Algorithm 2 TSAL Training

```
Step 1
    Require: labeled corpus X (source domain(s)); unlabeled corpus U(target domain)
    MAN training
Step 2
    Require: labeled corpus X (source domain(s)); unlabeled corpus U(target domain); $F_s$ from step1
    MAN training
```

	Elec.	dvd	Kitch.	apparel	camera	health	music	toys	video	baby	magaz.	softw.	sports	IMDb	MR	AVG.
MAN	69.9	80.35	71.0	75.3	72.1	74.8	81.3	79.3	79.85	72.85	74.0	76.35	73.3	81.25	71.0	75.51
TSAL	71.35	81.25	73.5	76.2	73.0	76.65	82.15	80.0	80.65	73.0	76.0	77.0	75.2	82.2	71.5	76.64

Table 3: Unsupervised domain adaptation results on the FDU-MTL review dataset. One source (books domain) V.S. one target domain.

The proposed TSAL training can be seen in Algorithm 2. In Step 1, the MAN's training is completed first. The learned shared extractor F_s is then used as an initialization for its peer in step 2, allowing it to learn more deeply.

Experiments

The efficiency of the proposed TSAL system is empirically evaluated in this section.

Datasets and Implementation Details

The first dataset on which we report our experiments is the Amazon reviews dataset (Blitzer, Dredze, and Pereira), which has been widely used for cross-domain sentiment classification. It contains pre-processed features that lose order detail, making it impossible to use powerful feature extractors (e.g. RNN or CNN). For a fair comparison, we use an MLP as our feature extractor and select the top 5000 features to represent each review as a 5000d feature vector, with feature values representing raw counts of the features. The Amazon dataset includes 2000 samples with binary labels for each of the four domains: books, DVDs, appliances, and kitchen products (positive, negative). we provide the mean accuracy as well as the standard errors over five runs, to illustrate the performance variance and conduct significance tests.

Also, the FDU-MTL dataset (Liu, Qiu, and Huang 2017) is used. Since Amazon reviews have already been tokenized and converted into a bag of features, the order data is lacking. As a result, we use the FDU-ML dataset, which contains raw description text and enables us to deploy effective feature extractor architectures (e.g. LSTM, CNN). In this dataset, there were 16 distinct domains. Books, appliances, DVDs, kitchen, clothes, cameras, fitness, music, toys, video, baby, magazine, games, and sports are among the 14 Amazon domains included, as are two movie review domains from the IMDb and MR datasets. The number of training and unlabeled data varies between 1400 and 2000 across domains. Besides, as a validation and test collection, each domain has 200 and 400 samples, respectively. In all of the experiments, the performance of our framework is measured by classification accuracy.

Experiments on the Amazon Dataset

To validate TSAL's effectiveness, we compare it to state-of-the-art multi-source domain adaptation methods. All baseline methods in the comparison include:

- **msDA (Chen et al. 2012)**: it employs marginalized stacked denoising auto-encoders to learn new representations for domain adaptation.
- **DANN (Ganin et al. 2016)**: it introduces the adversarial training as a representation learning approach for domain adaptation.
- **MDAN(H-MAX) and MDAN(S-MAX) (Zhao et al. 2017)**: they are two adversarial neural models.
- **MAN (Chen and Cardie 2018)**: it is a multinomial adversarial networks for multi-domain text classification.
- **DACL (Wu and Guo 2020)**: dual adversarial co-learning approach for text classification.

One of the four domains in the Amazon dataset is used as the target domain in four separate experiments, while the other three are used as source domains. On the left side of Table 1, the target domain is shown, and the test accuracy for TSAL and other multi-source domain adaptation is mentioned. for all four experiments, TSAL outperforms several baseline systems, such as an MLP trained on the source domains, as well as single-source domain adaptation methods such as mSDA (Chen et al. 2012), DANN (Ganin et al. 2016), MDAN (Zhao et al. 2017), and MAN (Chen and Cardie 2018). In all three experiments, TSAL outperforms DACL (Wu and Guo 2020), except for the one where the dvd is the target, in which Dacl outperforms TSAL by 0.2% (Table 1, second row). It's worth noting that TSAL outperforms other approaches on average. Furthermore, TSAL outperforms the MAN by 1.5 %, indicating that our proposed architecture's hypothesis is sound.

Experiments on the FDU-MTL

To gain more insights into the effectiveness of TSAL, we run a second set of experiments on the FDU-MTL dataset (Liu, Qiu, and Huang 2017). In the case of multiple sources and one target, Table 2 reports the test set accuracy for Man and TSAL. The target domain is shown on top, while the remaining 15 domains are the source domains. we attain TSAL success on every domain with an average overall accuracy of 85.94%, significantly outperforming the MAN by 2.65%. This finding confirms the validity of the TSAL hypothesis once more.

Moreover, Table 3 shows how well the proposed model performs with a single source and single target. By using books domain as a source domain, we show the average overall accuracy. As a source domain, we can use any domain other than books. As seen in Table 3, TSAL outperforms MAN for different domains. For example, when the target domain is the health domain, TSAL outperforms MAN by 2%. There is also a 2% rise when sports is used as a target domain. According to our results, TSAL improves the average overall accuracy by 1.5 % in single-source and single-target settings. For example, when the target domain is the health domain, TSAL outperforms MAN by 2%. There is also a 2% rise when sports is used as a target domain. According to our results, TSAL improves the average overall accuracy by 1.5 % in single-source and single-target problem.

Conclusion

In this paper, we introduce a two-step adversarial learning model for text classification in an unsupervised domain adaptation framework. TSAL includes two steps, analogous to fine-tuning, to boost domain-invariant feature representation learning. This is based on the premise that more deep features will be revealed,further assisting the target domain. The TSAL's performance is evaluated using two benchmark datasets: Amazon and FDU-MTL. Furthermore, our experiments include two settings: single source - single target domain and multi-source single target domain. The promising performance of TSAL is demonstrated by the experimental results. The frameworks proposed could be easily adapted to other text classification tasks or applied to several target domains.

References

Baktashmotlagh, M.; Harandi, M.; Lovell, B.; and Salzmann, M. 2013a. Unsupervised domain adaptation by domain invariant projection. In *Proceedings of the IEEE International Conference on Computer Vision*, 769–776.

Baktashmotlagh, M.; Harandi, M. T.; Lovell, B. C.; and Salzmann, M. 2013b. Unsupervised domain adaptation by domain invariant projection. In *Proceedings of the IEEE International Conference on Computer Vision*, 769–776.

Blitzer, J.; Dredze, M.; and Pereira, F. Domain adaptation for sentiment classification. In *45th Annv. Meeting of the Assoc. Computational Linguistics (ACL07).*

Blitzer, J.; McDonald, R.; and Pereira, F. 2006. Domain adaptation with structural correspondence learning. In *Proceedings of the 2006 conference on empirical methods in natural language processing*, 120–128. Association for Computational Linguistics.

Bousmalis, K.; Trigeorgis, G.; Silberman, N.; Krishnan, D.; and Erhan, D. 2016. Domain separation networks. In *Advances in neural information processing systems*, 343–351.

Chen, X., and Cardie, C. 2018. Multinomial adversarial networks for multi-domain text classification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 1226–1240.

Chen, M.; Xu, Z.; Weinberger, K. Q.; and Sha, F. 2012. Marginalized denoising autoencoders for domain adaptation. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, 1627–1634.

Das, D., and Lee, C. G. 2018. Sample-to-sample correspondence for unsupervised domain adaptation. *Engineering Applications of Artificial Intelligence* 73:80–91.

Daumé III, H. 2007. Frustratingly easy domain adaptation. *ACL 2007* 256.

Ganin, Y.; Ustinova, E.; Ajakan, H.; Germain, P.; Larochelle, H.; Laviolette, F.; Marchand, M.; and Lempitsky, V. Domain-adversarial training of neural networks.

Ganin, Y.; Ustinova, E.; Ajakan, H.; Germain, P.; Larochelle, H.; Laviolette, F.; Marchand, M.; and Lempitsky, V. 2016. Transfer hashing: From shallow to deep. *Journal of Machine Learning Research (JMLR)* 17(59):1–35.

Gong, B.; Shi, Y.; Sha, F.; and Grauman, K. 2012. Geodesic flow kernel for unsupervised domain adaptation. In *Proceedings of the 2012 IEEE conference on computer vision and pattern recognition*, 2066–73.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *Advances in neural information processing systems*, 2672–2680.

Gu, S.; Feng, Y.; ; and Liu, Q. 2019. Improving domain adaptation translation with domain invariant and specific information. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 3081–3091.

Kouw, W., and Loog, M. 2019. A review of single-source unsupervised domain adaptation. *arXiv preprint arXiv:1901.05335.*

Li, Z.; Zhang, Y.; Wei, Y.; Wu, Y.; and Yang, Q. 2017. End-to-end adversarial memory network for cross-domain sentiment classification. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2237–2243.

Liu, P.; Qiu, X.; and Huang, X.-J. 2017. Adversarial multi-task learning for text classification. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1–10.

Long, M.; Cao, Y.; Wang, J.; and Jordan, M. I. 2015. Learning transferable features with deep adaptation networks. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning-Volume 37*, 97–105.

Mathapati, S.; Nafeesa, A.; Tanuja, R.; Manjula, S.; and Venugopal, K. 2019. Semi-supervised domain adaptation and collaborative deep learning for dual sentiment analysis. *SN Applied Sciences* 1(8):907.

Quiñonero-Candela, J.; Sugiyama, M.; Schwaighofer, A.; and Lawrence, N. 2008. Covariate shift and local learning by distribution matching.

Schnabel, T., and Schutze, H. 2013. Towards robust cross-domain domain adaptation for part-of-speech tagging. In *Proceedings of the International Joint Conference on Natural Language Processing (IJCNLP)*, 198–206.

Tzeng, E.; Hoffman, J.; Zhang, N.; Saenko, K.; and Darrell, T. 2014. Deep domain confusion: Maximizing for domain invariance. *arXiv* arXiv–1412.

Tzeng, E.; Hoffman, J.; Saenko, K.; and Darrell, T. 2017. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7167–7176.

Vincent, P.; Larochelle, H.; Bengio, Y.; and Manzagol, P.-A. 2008. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, 1096–1103.

Wang, Y.; Huang, J.; Shang, J.; Niu, C.; and Zhou, Z. 2018. Domain invariant and class discriminative heterogeneous domain adaptation. In *2018 IEEE 3rd International Conference on Communication and Information Systems (ICCIS)*, 227–231. IEEE.

Wilson, G., and Cook, D. J. 2019. A survey of unsupervised deep domain adaptation. *arXiv preprint arXiv:1812.02849.*

Wu, Y., and Guo, Y. 2020. Dual adversarial co-learning for multi-domain text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 6438–6445.

Xiao, M., and Guo, Y. 2015. Semi-supervised subspace co-projection for multi-class heterogeneous domain adaptation. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 525–540. Springer.

Yosinski, J.; Clune, J.; Bengio, Y.; and Lipson, H. 2014. How transferable are features in deep neural networks? In *Advances in neural information processing systems*, 3320–3328.

Zhang, Y.; Xu, X.; and Hu, X. 2015. A common subspace construction method in cross-domain sentiment classification. In *Proceedings of the International Conference on Electronic Science and Automation Control (ESAC)*, 48 – 52.

Zhao, H.; Zhang, S.; Wu, G.; Costeira, J. P.; Moura, J. M.; and Gordon, G. J. 2017. Multiple source domain adaptation with adversarial training of neural networks. *arXiv* arXiv–1705.

Zhou, J.; Pan, S.; Tsang, I.; and Ho, S. 2016a. Transfer learning for cross-language text categorization through active correspondences construction. In *Proceedings of the (AAAI)*, 2400–2406.

Zhou, J.; Xu, X.; Pan, S.; Tsang, I.; Qin, Z.; and Goh, R. 2016b. Transfer hashing with privileged information. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2414–2420.

Zhou, J.; Zhao, H.; Peng, X.; Fang, M.; Qin, Z.; and Go, R. 2018. Transfer hashing: From shallow to deep. *IEEE Trans. Neural Netw. Learning Syst.* 29(12):6191–6201.