Published in final edited form as:
Proc Conf. 2019 June ; 2019: 414–419.

Simplified Neural Unsupervised Domain Adaptation

Timothy Miller
Computational Health Informatics Program, Boston Children's Hospital, Harvard Medical School

Abstract

Unsupervised domain adaptation (UDA) is the task of modifying a statistical model trained on labeled data from a source domain to achieve better performance on data from a target domain, with access to only unlabeled data in the target domain. Existing state-of-the-art UDA approaches use neural networks to learn representations that can predict the values of subset of important features called "pivot features." In this work, we show that it is possible to improve on these methods by jointly training the representation learner with the task learner, and examine the importance of existing pivot selection methods.

1 Introduction

Unsupervised domain adaptation (UDA) is the task of modifying a statistical model trained on labeled data from a source domain to achieve better performance on data from a target domain, without access to any labeled data in the target domain. Supervised domain adaptation methods can obtain excellent performance from a small number of labeled examples in the target domain (Daumé III, 2007), but UDA is attractive in cases where annotation requires specialized expertise or the number of meaningfully different sub-domains is large (e.g., both are true for clinical NLP).

Structural correspondence learning (Blitzer et al., 2006) (SCL) is one widely-used method for UDA in natural language processing. The key idea in SCL is that a subset of features, believed to be predictive across domains, are selected as *pivot features*. For each selected pivot feature, SCL creates an auxiliary classification task of predicting the value of that feature in an instance, given the values of all the non-pivot features for that instance. The auxiliary classifiers therefore learn important cross-domain information about the structure of the feature space, which the SCL algorithm uses to create an augmented representation that aligns features from different domains (further details in Section 2).

Meanwhile, recent advances in neural network learning have shown that training regimens that jointly consider evidence from multiple sources can improve performance – both multi-task learning (Søgaard and Goldberg, 2016) and fine tuning (Howard and Ruder, 2018; Devlin et al., 2018). However, existing SCL-based methods treat the representation learning and task learning as separate tasks, so the parameters of the representation learning machinery are fixed before training for the downstream task. Jointly learning the

timothy.miller@childrens.harvard.edu.

representation- and task-specific parameters can potentially allow a learning algorithm to find representations that are better suited for the task.

In this work, we describe a new UDA algorithm that is trained to jointly maximize two objectives: the primary supervised task in the source domain, and a pivot feature reconstruction task that can be trained on unlabeled data. We also explore the importance of pivot feature selection to this algorithm, in experiments that quantitatively and qualitatively examine the quality of existing pivot selection methods. We find that our joint neural approach to SCL improves unsupervised domain adaptation substantially on a standard sentiment classification task. Our results also show that while existing pivot selection methods per-form well, they are below an oracle-provided ceiling for many source-target pairs for the sentiment classification task we examine.

2 Background

This work builds off of existing work in unsupervised domain adaptation, starting with Blitzer's work on structural correspondence learning (SCL) (Blitzer et al., 2006, 2007). In the UDA task setup, one is given two datasets, the source $D_S = \{X_s, y_s\}$, with labels for each instance, and the target $D_T = \{X_t\}$, with unlabeled instances only. The goal of UDA is to learn a function $f_u(X_s, y_s, X_t)$ that improves on the classification performance over a function $f_l(X_s, y_s)$ when applied to new data drawn from the target distribution.

SCL is essentially a representation learning algorithm that works by creating a number of auxiliary classification tasks from the unlabeled source and target training instances (inspired by Ando and Zhang 2005). First, a set of *p pivot features* are selected, intended (in Blitzer's words) to be "features which behave in the same way for discriminative learning in both domains." Then, SCL creates p auxiliary tasks of predicting the value of pivot features in an instance given the non-pivot features in the instance. The weights of these linear classifiers are then aligned as columns in a matrix W, and the k left singular vectors are chosen from the singular value decomposition $W = U\Sigma V^{\top}$ to reduce its dimensionality, leading to a projection matrix $\theta \in R^{n \times k}$ that maps instances from the original feature space into the learned space. Most practical implementations find the best performance of SCL is obtained when projected features are concatenated with the original feature space; for some tasks and datasets other combinations have been tested and proved superior (Sapkota et al., 2016).

Recently, neural-network-based domain adaptation algorithms have been successful, including domain adversarial methods (Ganin et al., 2016) and auto-encoder-based methods (Glorot et al., 2011; Chen et al., 2014). However, a neural version of SCL still obtains near state-of-the-art performance (Ziser and Reichart, 2017). In that work, the AE-SCL system uses a multi-layer perceptron to replace the SVD for learning the feature projection. This network takes non-pivot features as input, has one hidden layer, and predicts the value of the pivot features at the output layer. Since it obtains supervision from the values of features, it can be trained on unlabeled instances from the source and target domains. To train for the down-stream sentiment classification task, the source in-stances are first passed into the trained representation learning network, and the values of the hidden layer are considered an additional set of features. These features are combined with all the original features, and the authors use a logistic regression classifier for the final sentiment classifier.

One standard corpus used to develop new domain adaptation algorithms is the Amazon sentiment analysis dataset.[1] This corpus was created by Blitzer et al. (2007), but we use the version included in the software release from Ziser and Reichart (2017)[2], along with their pre-processing steps, for ease of comparison with their results. This dataset contains reviews from four product categories on Amazon.com – books, DVDs, electronics, and kitchen appliances. Reviews are mapped to binary categories: *positive* if the review assigns the product >3 stars (out of 5) and *negative* if it assigns the product < 3 stars. This dataset also contains additional unlabeled instances for each category, used for training the pivot predictor.

3 Methods

The current work has two motivating factors. First, we would like to improve the performance of SCL using joint training. Existing SCL-based methods are successful in treating pivot prediction as a pre-training phase, but joint training may improve UDA by allowing the network to find representations that are equally good at pivot reconstruction but better for downstream task performance. Second, we would like to evaluate the quality of pivot selection methods and explore whether this step might be eliminated to simplify SCL.

We focus on feature-based UDA methods, as opposed to approaches that rely on embeddings (e.g., Barnes et al., 2018; Ziser and Reichart, 2018), since our primary interest is in improving existing models developed with a feature engineering approach. Such methods allow us to quickly adapt a number of different models to new datasets (e.g., for already-existing NLP pipeline software), rather than engineering new neural models from scratch for each of the pipeline tasks. For that reason, we compare to the AE-SCL model of Ziser and Reichart, rather than their subsequent models that take embeddings as input. In any case, we show that with some tuning the AE-SCL model can obtain state-of-the-art performance for many pairs.

3.1 Joint Neural Structural Correspondence Learning

Figure 1 graphically depicts our proposed joint model. The input to the model $\mathbf{x} \in \mathcal{R}^n$ is the set of all features extracted from the text – to compare with Ziser and Reichart (2017) we use unigrams and bigrams, extracted using scikitlearn (Pedregosa et al., 2011). We experimented with a few different hidden layer sizes, and settled on $d = 2000$ – this balances the need of the representation to predict more output variables than the AE-SCL method with run-time constraints. The representation is generated with $h(\mathrm{x}) = ReLU(W_h\mathrm{x})$, for $W_h \in \mathcal{R}^{d \times n}$. The task prediction is $f_{task}(\mathrm{x}) = Sigmoid\left(W_t h(\mathrm{x})\right)(W_t \in \mathcal{R}^{1 \times d})$ and the pivot prediction is $f_{pivot}(\mathrm{x}) = \left(W_p h(\mathbf{x})\right)\left(W_p \in \mathcal{R}^{p \times d}\right)$.

[1] https://www.cs.jhu.edu/~mdredze/datasets/sentiment/

[2] https://github.com/yftah89/Neural-SCL-Domain-Adaptation

The joint loss function for labeled source data D_l, all data D_a, and model parameters θ is:

$$\mathscr{L}(D;\theta) = \sum_{\mathbf{x}, y \in D_l} BCE\left(f_{task}(\mathbf{x}), y\right) + \lambda \sum_{\mathbf{x} \in D_a} BCE\left(f_{pivot}(\mathbf{x}), pivots(\mathbf{x})\right) + \rho\mathscr{R}(\theta) \tag{1}$$

where BCE is the binary cross-entropy loss, λ controls the weight of pivot prediction loss, *pivots* is a function that selects the indices from an instance that are the pivot features to be predicted, and ρ is the weight of the regularization term $\mathscr{R}$.

To train this model, we alternate passing labeled source data and unlabeled data from the source and target domains into the network. For the labeled data, the error term is the sum of task- and pivot-prediction tasks, while for the unlabeled data only the pivot-prediction loss is computed.

Training proceeds for 30 epochs, with mini-batch size of 50 instances, using the Adam optimizer (Kingma and Ba, 2014) with learning rate 0.001. For the loss function weight, we used $\rho = 0.1$ and λ = 100. We used held-out source data to compute validation loss after each epoch and selected the trained model with the lowest validation loss.

3.2 Pivot Selection for Neural Systems

One standard way of choosing pivot features is by calculating mutual information (MI) between the source features and labels, and selecting the features with the highest MI. It is far from clear, however, that this technique is always optimal. Earlier experiments with the POS tagging task (Blitzer et al., 2006) used feature frequency instead, and the extent of the correlation between frequency and MI for that task is not established.

Here, we attempt to provide some evidence about the quality of MI for the task of sentiment classification, using the classification pair of books to electronics. First, we wanted to rule out the null hypothesis that prediction of MI pivots is essentially a generic representation learning algorithm – in other words, that a network learning structure between any sets of sufficiently common features may improve adaptation performance. We modified Ziser et al.'s code to simply select random feature indices from the subset of those that occurred frequently enough to be pivot candidates. With this setup, adaptation performance averaged 0.724 across ten runs, well below their reported 0.744, casting doubt on the null hypothesis.

Next, we want to examine the contention that features with high MI relative to source labels are general. To do this, we simply compare the list of MI features used when the source is *books* to those when the source is *electronics*. We find that, out of the 100 pivot features selected by MI in either cases, there is overlap of 26 features, some examples of which are shown in Table 1 (left). Table 1 also shows a number of MI-selected pivots from the *books* domain that are not general (middle), and then a set of features MI-correlated with the target domain that seem general (right). These latter two columns are essentially precision and recall errors of the MI pivot selection algorithm.

Finally, we perform an oracle-based adaptation experiment, where we select the pivot feature indices using MI against the gold labels of the *target* domain, but then proceed with training looking only at source labels, with results in Table 2 discussed below.

4 Evaluation

We follow the standard setup for the Amazon sentiment task, splitting each source dataset into 1600 training and 400 validation instances, and evaluating on the entire labeled target dataset for each pair. We compare against two baselines: First, the reported results of Ziser and Reichart, and second, our replication of their results using their code. Our replication changed their code by replacing the stochastic gradient descent optimizer with Adam (Kingma and Ba, 2014), and increasing the training batch size from 1 to 50. These changes were made to speed training runs during development; we found they produced better-than-reported results and include these superior results as an even stronger baseline. We report results of two configurations of our joint learner. The first configuration ($Joint_{MI}$) uses the MI between source labels and features to select 100 pivot features. The second configuration ($Joint_{Oracle}$) is an oracle-informed system where we use the MI between *target* labels and features to select pivot features, but only use source labels while training the network. Both the AE-SCL$_R$ model and our $Joint_{MI}$ model were run for 10 iterations to minimize differences due to random initialization and to calculate significance statistics.

Table 2 shows the results of our experiments. First, we note that our replication of AE-SCL improves upon their reported results in 8 of 12 pairs, often by substantial margins, and is only worse in one pair (Kitchen→Books). Our $Joint_{MI}$ method is superior to the reported AE-SCL results in all pairs, 1.7 points (absolute) on average, and significantly better than the AE-SCL$_R$ in 9 of 12 pairs, using Welch's one-tailed t-test. This is, to our knowledge, the best result on this task using a feature-based approach (i.e., excluding systems that use embeddings). Despite constraining our system to adapting feature-based models, this result is competitive with the best-known result using a pure neural approach with embeddings as input, as Ziser and Reichart (2018) report an average accuracy of 0.804. The $Joint_{Oracle}$ configuration shows that, despite the large gains of joint training, there is still significant improvement available with better pivot selection.

5 Discussion and Conclusion

Our results show that by jointly learning representations and task networks, UDA can be greatly improved over existing neural UDA methods. We note that there are existing domain adaptation methods that use joint training with auxiliary tasks. Yu and Jiang (2016) use an auxiliary task of predicting whether a masked pivot word in a sentence is positive or negative sentiment, where they introduce a new technique to select pivots that still is based on correlations with source labels. Our work is unique in showing that the standard task of mutual-information-selected pivot prediction is a high quality auxiliary task, though future work should explore whether their pivot selection algorithm is superior to MI in our joint model. We also showed that existing neural UDA methods can be improved significantly with minor changes to the training regimen. Finally, we show that mutual information pivot selection is quite far from the performance ceiling provided by oracle-based pivot selection.

This work evaluated on the widely-used Amazon sentiment dataset from Blitzer et al. (2007). However, we believe that future work on domain adaptation should phase out the use of this dataset.[3] The test set for this setup is flawed in two important ways: first, it is artificially balanced with positive and negative reviews, when the problem is not actually balanced; it also has 3-star reviews removed, which is not a realistic test set setup without looking at labels. For these reasons, we recommend that future work use different domain adaptation datasets.

The Pytorch implementation used to produce these results is publicly available.[4]

Acknowledgments

Research reported in this publication was supported by the National Library Of Medicine of the National Institutes of Health under Award Number R01LM012918. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

References

Kubota Ando Rie and Zhang Tong. 2005 A framework for learning predictive structures from multiple tasks and unlabeled data. Journal of Machine Learning Research, 6(Nov):1817–1853.

Barnes Jeremy, Klinger Roman, and Walde Sabine im. 2018 Projecting Embeddings for Domain Adaptation: Joint Modeling of Sentiment in Diverse Domains In Proceedings of COLING 2018, the 27th International Conference on Computational Linguistics, Santa Fe, USA Association for Computational Linguistics.

Blitzer J, Dredze M, and Pereira F. 2007 Biographies, Bollywood, Boom-boxes and Blenders: Domain Adaptation for Sentiment Classification. In ACL 2007, page 440.

Blitzer J, McDonald R, and Pereira F. 2006 Domain adaptation with structural correspondence learning. Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP 2006).

Chen Minmin, Weinberger Kilian, Sha Fei, and Bengio Yoshua. 2014 Marginalized denoising auto-encoders for nonlinear representations. In International Conference on Machine Learning, pages 1476–1484.

Daumé Hal III. 2007 Frustratingly Easy Domain Adaptation. In Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, pages 256–263.

Devlin Jacob, Chang Ming-Wei, Lee Kenton, and Toutanova Kristina. 2018 Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

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

Glorot Xavier, Bordes Antoine, and Bengio Yoshua. 2011 Domain adaptation for large-scale sentiment classification: A deep learning approach. In Proceedings of the 28th international conference on machine learning (ICML-11), pages 513–520.

Howard Jeremy and Ruder Sebastian. 2018 Universal language model fine-tuning for text classification. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), volume 1, pages 328–339.

Kingma Diederik P and Lei Ba Jimmy. 2014 Adam: Amethod for stochastic optimization. In Proc. 3rd Int. Conf. Learn. Representations

Pedregosa Fabian, Varoquaux Gal, Gramfort Alexandre, Michel Vincent, Thirion Bertrand, Grisel Olivier, Blondel Mathieu, Prettenhofer Peter, Weiss Ron, Dubourg Vincent, and others. 2011

[3] Thanks to the anonymous reviewer who made this argument which we found convincing. We credit them with these points while accepting the blame for any poor communication of these points.

[4] https://github.com/tmills/Neural-SCL-Domain-Adaptation

Scikitlearn: Machine learning in Python. Journal of machine learning research, 12(Oct):2825–2830.

Sapkota U, Solorio T, Gómez M Montes-y, and Bethard S. 2016 Domain Adaptation for Authorship Attribution: Improved Structural Correspondence Learning. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pages 2226–2235.

Søgaard A and Goldberg Y. 2016 Deep multi-task learning with low level tasks supervised at lower layers. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pages 231–235.

Yu Jianfei and Jiang Jing. 2016 Learning sentence embeddings with auxiliary tasks for cross-domain sentiment classification. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 236–246.

Ziser Yftah and Reichart Roi. 2017 Neural Structural Correspondence Learning for Domain Adaptation. In Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017), pages 400–410. Association for Computational Linguistics.

Ziser Yftah and Reichart Roi. 2018 Pivot Based Language Modeling for Improved Neural Domain Adaptation. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pages 1241– 1251. Association for Computational Linguistics.

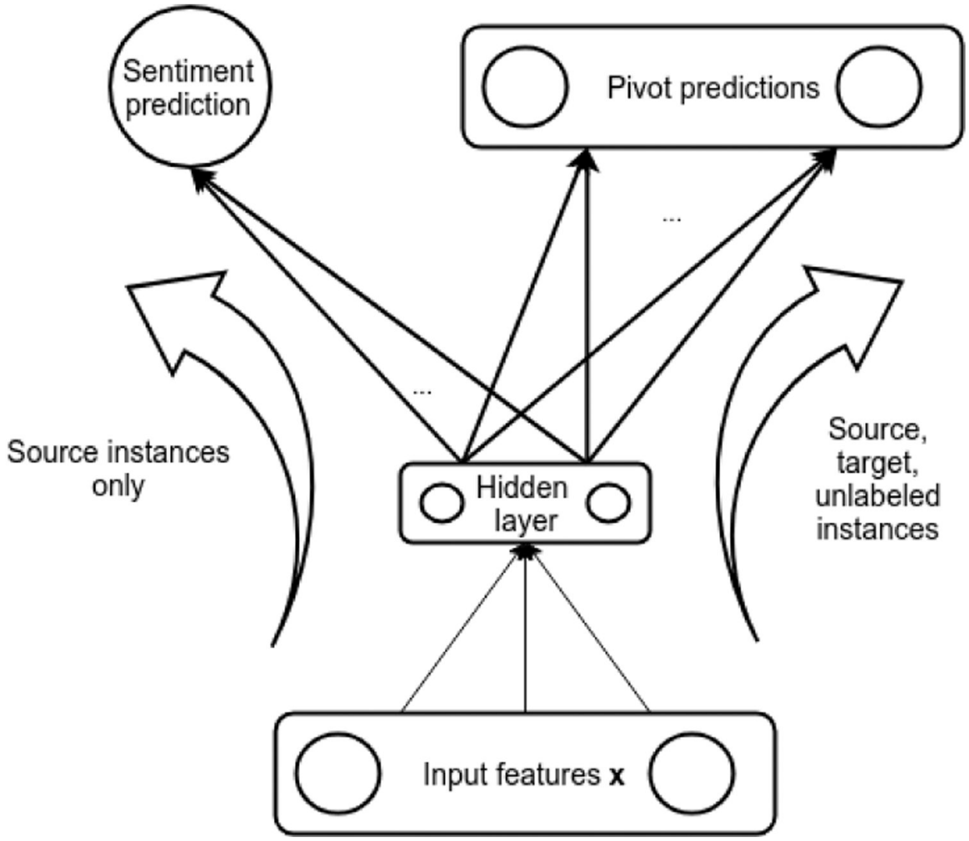

Figure 1:
Network architecture for unsupervised domain adaptation.

Table 1:

Example features selected as pivots using MI. *Shared features* indicates those selected for both domains. *Book (spec)* highlights pivots selected in the books domain that appear domain-specific. *Elec (gen)* indicates high-MI pivots in the electronics domain that appear domain-general that are not selected when *books* is the source domain.

Shared features	Book (spec)	Elec (gen)
an excellent	Writing	and great
best	care about	is perfect
bad	flat	never buy
highly recommend	pages	not good
waste your	finish	poor

Table 2:

Summary of results. B=Books, D=Dvd, E=Electronics,K=Kitchen. AE-SCL=Reported results from Ziser and Reichart (2017). AE-SCL$_R$=Replicated results from the same. $Joint_{MI}$=results from the joint model in Section 3.1. $Joint_O$=results from the joint model using oracle MI pivot selection. Bold indicates a significant difference ($p < 0.05$) between AE-SCL$_R$ and $Joint_{MI}$ using Welch's one-tailed t-test.

S→T	AE-SCL	AE-SCL$_R$	$Joint_{MI}$	$Joint_O$
B→D	0.794	**0.812**	0.808	0.84
B→E	0.744	0.761	**0.786**	0.818
B→K	0.795	**0.814**	0.809	0.831
D→B	0.758	0.779	**0.807**	0.808
D→E	0.763	0.763	**0.803**	0.843
D→K	0.8	0.821	**0.83**	0.832
E→B	0.701	0.701	**0.738**	0.749
E→D	0.732	0.748	**0.766**	0.824
E→K	0.848	0.848	**0.875**	0.869
K→B	0.742	0.731	**0.744**	0.767
K→D	0.743	0.754	0.752	0.828
K→E	0.828	0.841	**0.856**	0.851
Ave.	0.771	0.781	**0.798**	0.821