Unsupervised Domain Adaptation for Sentiment Classification using Pseudo-Labels

Ruishan Liu *
Department of Electrical Engineering
Stanford University
Stanford, CA 94305
ruishan@stanford.edu

Liyue Shen *
Department of Electrical Engineering
Stanford University
Stanford, CA 94305
liyues@stanford.edu

Abstract

With fast increase of online recommendations and reviews, labeling efficient data in thousands of domains for natural language processing (NLP) is not feasible in practice. As an important category of domain adaptation methods, pseudo-labeling combined with deep neural network remains absent for those NLP tasks in literature. Especially in the field of sentiment classification, little is understood about the properties of pseudo label selection. How does the selected pseudo labels affect the learning performance? How much can we trust the pseudo labels? Motivated by both research and practical needs, we propose several end-to-end deep learning frameworks to tackle the domain adaptation problem in natural language processing. We learn to transfer a sentiment classifier trained on source domain with sentiment annotations to target domain without any label. We qualitatively examine how the selection rule affects the classification performance and evaluate the pseudo-label related approaches.

1 Introduction

With the rapid growth of social media such as online reviews and ratings, a large variety of natural language data become available, which enables tremendous applications. The numerous data sources raise one problem — how to learn cross domain problems robustly and generally. Among all the machine learning tasks, sentiment classification is considered as an important benchmark in academia and a critical application in industry. The cross domain problem becomes extremely severe for sentiment classification, which usually spans a large number of domains. For example, reviews on *kitchen* may use description such as "blunt", "delicious" and "soft", while reviews on *book* are more likely to include "profound", "concise" and "coherent". The different data distributions make it hard to develop a general sentiment classifier.

The most advance progress in domain adaptation has been achieved in the field of visual recognition. Among recently developed methods, giving pseudo-labels to unlabeled target samples has become an important category as it reaches state-of-the-art performance on digit recognition (Sener et al. (2016); Saito et al. (2017)). However, current sentiment domain adaptation mainly focuses on representation based methods, and the pseudo-labeling related approach has not been adopted in literature.

2 Related Work.

The problem of unsupervised domain adaptation for sentiment classification in natural language processing has been addressed by different approaches such as learning intermediate representation

*These authors contributed equally to this work and are listed in alphabetical order.

(Glorot et al. (2011); Deng et al. (2014)), active learning (Wu et al. (2017)) and adversarial training (Ganin et al. (2016)).

Referring to the similar problem in the field of computer vision applications, domain adaptation has achieved impressive performances combined with deep learning (Long et al. (2015)). Especially, pseudo-label related methods are leveraged to transfer representations across domain gap (Sener et al. (2016); Saito et al. (2017)). On the other hand, in order to solve the inefficient labeling problem for semi-supervised learning research, temporal ensembling method (Laine & Aila (2017)), mean-teacher model weights ensembling method (Tarvainen & Valpola (2017)) as well as the approach of virtual adversarial training (Takeru Miyato & Ishii (2016)) have achieved impressive performance on MNIST, CIFAR and ImageNet datasets.

Based on the related works of domain adaptation and semi-supervised learning in computer vision tasks, we propose our methods to solve the unsupervised domain adaptation problem in natural language processing in this paper.

3 Approach

Vanilla Baseline. Motivated by the comparative study of different deep model architecture in natural language processing in Yin et al. (2017), we propose to explore various deep networks for our sentiment analysis task such as RNN, CNN and LSTM. In our baseline, we test the performance of both RNN and LSTM network. For the vanilla domain task, we train an RNN/ LSTM network based on source samples, and then evaluate model performance on target domain. Note that there is no training data and labels from target domain for training the network.

3.1 Hard Pseudo-Labeling based Methods.

One simple but important way to deal with unlabeled data is pseudo-labeling. That is, we leverage the currently model trained on annotated source examples to assign fake labels to the targets examples, which are then added into training dataset through iteratively update. In this sense, we propose a straightforward pseudo-labeling method and also labeling with tri-training mechanism.

Proposed Method 1: Pseudo-labeling. Consider the particular knowledge transfer for domain adaptation, we implement a multi-model learning architecture leveraging pseudo-labeling approach. Specifically, in order to leverage the unlabeled data in the target domain, we use a training model to evaluate unannotated target samples and assign pseudo labels in each training epoch. In addition, a hyper-parameter is set here as the threshold to determine how many pseudo-labeled target samples should be updated into the total training data set for every epoch. Finally, we test the developed model on the target domain.

Proposed Method 2: Tri-training with Pseudo-labels. Motivated by the paper Saito et al. (2017), in order to improve the accuracy of predicting pseudo-labels to unlabeled target samples, we train two classifier simultaneously to work as the predictor. When we assign pseudo labels to target samples in each training epoch, a new target sample will be chosen to add into training dataset only if two criteria are satisfied: 1) The two classifiers predict the same label; 2) Both of the two predictors achieve a confidence score exceeding the threshold. Such more strict constraints improve the process of pseudo-labeling by selecting predicted target samples with higher confidence. The different initializations of two classifiers also avoid the affect resulted from the randomness in training classifier on source labeled data. Therefore, tri-training will work better as an advanced pseudo-labeling approach.

3.2 Loose Pseudo-Labeling based Methods.

The pseudo-labeling based methods introduced above directly assign the pseudo category labels to the training examples, and the targets are hard one-hot labels which will lose the information in probability distribution predicted from current model. In order to solve this problem, we propose new methods based consistency loss. In other words, instead of assign hard pseudo labels to training samples, the algorithm uses the predicted confidence score as the soft target, which is integrated as the consistency loss into the total objective functions for training the whole network.

Algorithm 1 Tri-training with Pseudo-labels.

Require: Source data $\mathcal{X}_{\text{source}}$, Source label $\mathcal{Y}_{\text{source}}$, Target data $\mathcal{X}_{\text{target}}$
1: initial training set $\mathcal{X}_{\text{label}} := \mathcal{X}_{\text{source}}$, $\mathcal{Y}_{\text{label}} := \mathcal{Y}_{\text{source}}$, $\mathcal{T} = (\mathcal{X}_{\text{label}}, \mathcal{Y}_{\text{label}})$;
2: **for** $i = 1$ to n **do**
3: Train $Model_A$ and evaluate on target data: $(\hat{\mathcal{Y}}_{\text{A}}, \mathcal{S}_{\text{A}})$:= $Model_A$ ($\mathcal{T}$, $\mathcal{X}_{\text{target}}$)
4: Train $Model_B$ and evaluate on target data: $(\hat{\mathcal{Y}}_{\text{B}}, \mathcal{S}_{\text{B}})$:= $Model_B$ ($\mathcal{T}$, $\mathcal{X}_{\text{target}}$)
5: Select target samples from two evaluation results and assign pseudo-labels: $(\mathcal{X}_{select}, \hat{\mathcal{Y}}_{pred}, \mathcal{S}_{pred})$:= findmax($\mathcal{X}_{\text{target}}, \hat{\mathcal{Y}}_{\text{A}}, \mathcal{S}_{\text{A}}, \hat{\mathcal{Y}}_{\text{B}}, \mathcal{S}_{\text{B}}$)
6: Update training dataset: $\mathcal{X}_{\text{label}} := \mathcal{X}_{\text{label}} \bigcup \mathcal{X}_{select}$, $\mathcal{Y}_{\text{label}} := \mathcal{Y}_{\text{label}} \bigcup \hat{\mathcal{Y}}_{pred}$.
7: **end for**

Proposed Method 3: Temporal-ensembling. In order to get a high-accuracy probability distribution for the consistency loss, we need a more reliable predictor to serve as the target scores. Motivated by the semi-supervised methods of temporal ensembling Laine & Aila (2017) and mean-teacher modeling Tarvainen & Valpola (2017), temporal model ensemble is an efficient approach to predict more reliable targets compared to the training model at current epoch, since the overfitting along with the training process. Base on this observation, we propose a temporal ensembling method to predict targets for unlabeled training samples on target domain. To be specific, the prediction from an ensembling model is computed as the moving average values at each epoch. Then the averaged confidence scores are leveraged to update consistency loss function for the training process on target samples. Thus, the objective loss function for training processes are defined as the weighted combination of classification loss on source data and consistency loss on target data as follows.

$$L_{objective} = w_1 \times L_{classification} + w_2 \times L_{consistency} \tag{1}$$

Where w_1 and w_2 are loss weights. Classification loss is defined as binary cross-entropy loss function for source examples and labels. Consistency loss is defined as mean-square-error loss function for target examples and predicted pseudo labels of probability distribution.

Proposed Method 4: Tri-training with Temporal-ensemble. Similar as Method 2, tri-training is an effective mechanism for unsupervised domain adaptation methods supported by results in Saito et al. (2017). We also add the tri-training mechanism on the top of temporal-ensembling with consistency loss for this task of unsupervised domain adaptation. Specially, the mean values of predicted confidence scores from two classifiers are set as the targets for consistency loss function. Similarly, when we iteratively train the model based on target domain, the target examples will be chosen into training set only if these two criteria are satisfied: 1) The two classifiers predict the same label; 2)

Table 1: Results of our two proposed hard pseudo-label related methods (Pseudo-labeling and Tri-training) with RNN. Here we show the test accuracy on the target domain. The vanilla approach is baseline for comparison.

Source→Target	Valinna	Pseudo-Label	Tri-Training
books→dvd	**55.9 %**	53.8 %	54.1 %
books→electronics	51.5 %	**56.3 %**	51.7
books→kitchen	53.1 %	**56.1 %**	54.5 %
dvd→books	52.8 %	52.2 %	**53.5 %**
dvd→electronics	51.7 %	**52.8 %**	52.1 %
dvd→kitchen	52.6 %	53.0 %	**58.0 %**
electronics→books	52.4 %	53.1 %	**56.4 %**
electronics→dvd	53.3 %	52.3 %	**53.5 %**
electronics→kitchen	**56.5 %**	55.8 %	56.2 %
kitchen→books	**56.0 %**	51.5 %	53.5 %
kitchen→dvd	52.0 %	52.4 %	**54.0 %**
kitchen→electronics	**56.7 %**	53.7 %	55.0 %

Table 2: Results of our proposed methods ("PL", "Tri", "TE" stand for Pseudo-Labeling, Tri-training and Temporal-Ensembling respectively) with two neural network architectures (RNN and LSTM). Here we show the test accuracy on the target domain, while different methods are trained with different data source. The vanilla approach trained with only source data is the baseline for comparison. (S+T) notes that the model is trained on both labeled source data and pseudo-labeled target data, while (T) means training with only pseudo-labeled target data.

Source→Target	**RNN**	**LSTM**					
	Tri	**Valinna**	**PL**	**Tri+PL**		**TE**	
	(S+T)	**(S)**	**(S+T)**	**(S+T)**	**(T)**	**(S+T)**	**(T)**
books→dvd	54.1 %	68.5 %	**69.8** %	67.6 %	64.9 %	64.2 %	64.6 %
books→electronics	51.7 %	65.4 %	**66.1** %	62.8 %	51.9 %	64.1 %	64.4 %
books→kitchen	54.5 %	69.6 %	68.2 %	**69.0** %	66.5 %	63.9 %	68.5 %
dvd→books	53.5 %	69.0 %	**68.3** %	68.2 %	65.3 %	62.4 %	65.2 %
dvd→electronics	52.1 %	66.6 %	**66.1** %	62.3 %	50.5 %	57.9 %	60.9 %
dvd→kitchen	58.0 %	66.9 %	**66.4** %	64.7 %	54.8 %	61.3 %	63.8 %
electronics→books	56.4 %	66.4 %	65.9 %	**66.8** %	64.4 %	60.3 %	64.0 %
electronics→dvd	53.5 %	65.6 %	**63.8** %	62.3 %	54.8 %	59.0 %	61.4 %
electronics→kitchen	56.2 %	74.4 %	**73.6** %	73.5 %	68.5 %	68.3 %	62.1 %
kitchen→books	53.5 %	66.6 %	**66.9** %	65.2 %	65.1 %	63.9 %	69.9 %
kitchen→dvd	54.0 %	65.9 %	**64.0** %	61.2 %	53.0 %	61.1 %	63.8 %
kitchen→electronics	55.0 %	71.2 %	69.6 %	**71.2** %	71.8 %	63.9 %	68.9 %

Both of the two predictors achieve a confidence score exceeding the threshold. In such a sense, the target samples with higher reliability are chosen for training while target examples with low prediction confidence are filtered out.

4 Experiments

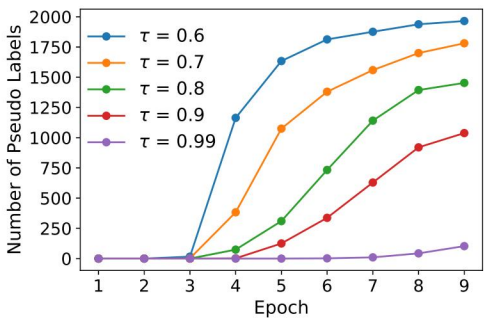
We carried out experiments for our four proposed methods with two deep neural network architectures (RNN and LSTM) on a benchmark multi-domain sentiment dataset.

Dataset. Amazon product review dataset is chosen as a benchmark sentiment domain adaptation dataset. This dataset (He & McAuley (2016)) contains 142.8 million product reviews from Amazon in 25 various product categories, while users' rating scores are regarded as the target labels in the task of sentiment classification. In addition, because of richness of specific product domains, this Amazon review dataset is widely used for evaluation in previous works related to domain adaptation tasks Glorot et al. (2011); Chen et al. (2012); Ganin et al. (2016); Saito et al. (2017). Considering the results comparison with these previous works, we follow the same data pre-processing and experimental setting. In particular, 12 domain adaptation scenarios are explored in our problem which are constructed from 4 amazon review dataset: books, dvd, electronics and kitchen.

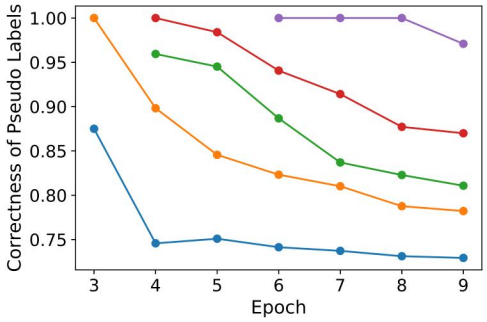
Data Pre-processing. Following the experimental setting in Saito et al. (2017), we pre-process the dataset by regarding start 1 - 3 as negative class, while start 4 and 5 as positive for the sentiment task. 2000 labeled source samples and 2000 unlabeled target samples are generated from four categories including "books", "dvd", "electronics" and "kitchen". Then 12 domain adaptation tasks are studied, such as "dvd" → "book" where "dvd" is the source domain and "book" is the target one. Both training and test data sets are balanced with equal number of positive and negative samples. Another 4000 samples are used for model evaluation on target domain.

Word Embedding: In order to do word embedding, we propose to leverage the word2vec method for text representations in our problem. 1024-dimension word vectors are trained respectively with training set using word2vec approach.

Experiment Setup. We implement our network using Keras library. The network structure contains on one embedding layer, one hidden layer of RNN or LSTM, and one output layer with softmax



(a) Number of total adopted pseudo labels after each training epoch.



(b) The correctness of the assigned labels of pseudo-labeled data.

Figure 1: Learning curve for pseudo label generation with different selection threshold τ in Tri-training. Here the source domain is electronics and target domain is kitchen.

activation. In order to train the networks, we use binary cross entropy loss function and Adam optimizer, with 32 samples per mini-batch and in total 10 epochs.

Evaluation Methodology. In order to evaluate the transferability of the model across different domains, we use the sentiment classification task as evaluation experiments, and the classification accuracy as evaluation metrics. Under the sentiment recognition task, we train our model on labeled data in source domain and unlabeled data in target domain. An accurate sentiment classifier on target domain is what we expect.

5 Results

5.1 Performance.

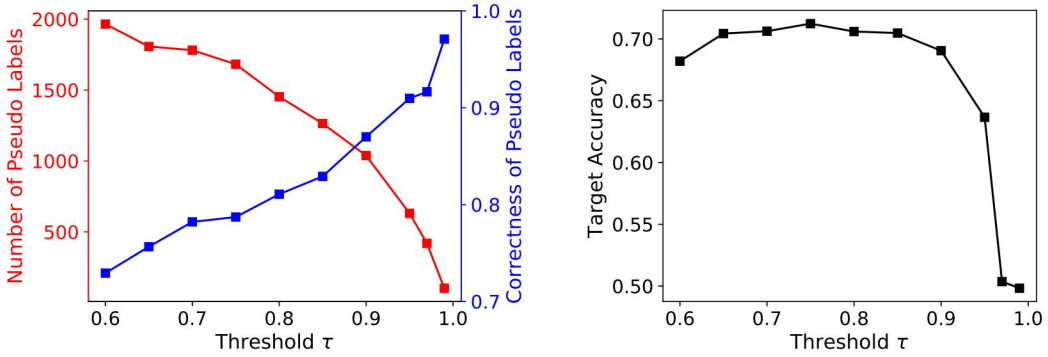
The two architectures, RNN and LSTM, exhibit different performances with the pseudo-label related methods. The test accuracies on the target domain for vanilla and our proposed methods are given in Table 1 (RNN) and Table 2 (LSTM).

Network Architectures. In the experiment, we test our methods with both RNN and LSTM. Overall, LSTM has a better performance over RNN. For example, when source is books and target is electronics, the best target test accuracy achieved by RNN is 56.3 %, while the best accuracy is 69.8 % for LSTM. Meanwhile, the pseudo label related methods also have different performance for the two architectures.

Pseudo Label Methods Evaluation. For RNN, tri-training (Tri) performs slightly better than pseudo-labeling (PT), while PT performs best over all methods for LSTM. First, for RNN, table 1 shows improved performance of both pseudo-label and tri-training methods. For example, the tri-training method has the potential to improve the accuracy by 5.4 % for "dvd→electronics" case. In the meantime, the contribution of LSTM is not significant, as indicated by Table 2. We note that to this point, we only do the fine-tuning on hyper-parameters for vanilla setting. Further improvement is expected after hyper-parameter search and architecture tuning are done particularly for tri-training. The loose pseudo-label methods, i.e., Temporal-ensembling, does not perform well overall.

5.2 Pseudo-label Generation.

The assignment of pseudo labels is the core of all our methods. To this point, the properties of pseudo label selection are not revealed. How does the selected pseudo labels affect the learning performance? How much can we trust those pseudo labels? Here we qualitatively analyze how the selection affects the classification performance, with tri-training as an illustration.



(a) Total adopted pseudo label number (red, left axis) and correctness of assigned labels (blue, right axis).

(b) Target test accuracy when trained on pseudo-labeled data.

Figure 2: Performance for different selection threshold τ in Tri-training. The results are retrieved at the end of the training, corresponding to the last epoch in Fig. 1. Here the source domain is electronics and target domain is kitchen.

Generation of Pseudo Labels. The trends of pseudo-label generation are illustrated in Fig. 1a. The pseudo labels begin to generate at a few epochs, and the number of assigned labels first increases rapidly and gradually saturates. For example, when selection threshold τ is 0.9, the tri-training algorithm starts to accept pseudo labels at epoch 3, and the pseudo label number reaches 1000 at the end of the training. A looser principle, i.e., a smaller confidence threshold τ, leads to a larger pseudo label set as expected.

We further examine the quality of generated pseudo labels, as shown in Fig. 1b. Here the quality is evaluated by correctness — the proportion of correct assigned pseudo labels. The correctness is found to gradually decrease during the training.

Effects of Selection Rule. The influence of pseudo label selection arises from the trade-off between the quantity and quality of pseudo labels. As the selection rule becomes more strict, the quantity increases monotonically and the quality decreases monotonically. That is, a higher confidence threshold τ results in a high-quality but smaller pseudo label set, as indicated by Fig. 2a. When trained on pseudo-labeled data, the target test accuracy is affected by the selection threshold τ following Fig. 2b. For example, a very large threshold $\tau = 0.99$ suffers from the limited pseudo label number around 100, while a small threshold $\tau = 0.6$ is hindered by the low correctness about 0.7.

6 Discussion and Conclusion

In our experiments, we evaluate four pseudo-label related methods with two network architectures in sentiment domain adaptation tasks, which have good performance and much potential. The influence of pseudo label selection is found to arise from the trade-off between the quantity and quality of pseudo labels. We note that there is much space for us to improve the current performance and more analysis could be carried out.

Further Improvement. In our current experiments,we carefully fine-tuned the hyper-parameters and network architecture for vanilla setting. Good improvement is made over the result in Milestone. However, the tri-training setting differs a lot from the plain vanilla one. Further fine-tuning should be made particularly for tri-training case, and improvements are expected. Moreover, we also do not deal with the problem of overfitting. The training accuracies could always go to 100%, with the test accuracy given in Table 2. We plan to try different approaches such as adding regularization term and conducting early-stopping with validation datasets.

Exploration of Different Deep Models. As shown in our work, different neural network architectures could have different performance for sentiment domain adaptation tasks and different char-

acteristic could be perceived. Here we tested on RNN and LSTM, and our study could be extended to other deep models such as GRU and CNN.

References

Chen, Minmin, Xu, Zhixiang, Weinberger, Kilian Q., and Sha, Fei. Marginalized denoising autoencoders for domain adaptation. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, ICML'12, pp. 1627–1634, USA, 2012. Omnipress. ISBN 978-1-4503-1285-1.

Deng, Jun, Zhang, Zixing, Eyben, Florian, and Schuller, Björn. Autoencoder-based unsupervised domain adaptation for speech emotion recognition. *IEEE Signal Processing Letters*, 21(9):1068–1072, 2014.

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

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

He, Ruining and McAuley, Julian. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *proceedings of the 25th international conference on world wide web*, pp. 507–517. International World Wide Web Conferences Steering Committee, 2016.

Laine, Samuli and Aila, Timo. Temporal ensembling for semi-supervised learning. *ICLR*, 2017.

Long, Mingsheng, Cao, Yue, Wang, Jianmin, and Jordan, Michael. Learning transferable features with deep adaptation networks. In *International Conference on Machine Learning*, pp. 97–105, 2015.

Saito, Kuniaki, Ushiku, Yoshitaka, and Harada, Tatsuya. Asymmetric tri-training for unsupervised domain adaptation. *arXiv preprint arXiv:1702.08400*, 2017.

Sener, Ozan, Song, Hyun Oh, Saxena, Ashutosh, and Savarese, Silvio. Learning transferrable representations for unsupervised domain adaptation. In *Advances in Neural Information Processing Systems*, pp. 2110–2118, 2016.

Takeru Miyato, Shin-ichi Maeda, Masanori Koyama and Ishii, Shin. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *ICLR*, 2016.

Tarvainen, Antti and Valpola, Harri. Weight-averaged consistency targets improve semi- supervised deep learning results. *NIPS*, 2017.

Wu, Fangzhao, Huang, Yongfeng, and Yan, Jun. Active sentiment domain adaptation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pp. 1701–1711, 2017.

Yin, Wenpeng, Kann, Katharina, Yu, Mo, and Schütze, Hinrich. Comparative study of cnn and rnn for natural language processing. *arXiv preprint arXiv:1702.01923*, 2017.