Self-Adapter at SemEval-2021 Task 10: Entropy-based Pseudo-Labeler for Source-free Domain Adaptation

Sangwon Yoon[1,2], Yanghoon Kim[1,3] and Kyomin Jung[1,3]
[1]Seoul National University, Seoul, Korea
[2]School of Law, Seoul National University, Seoul Korea
[2]Department of Electrical and Computer engineering, Seoul National University, Seoul Korea
{sangwon38383, ad26kr, kjung}@snu.ac.kr

Abstract

Source-free domain adaptation is an emerging line of work in deep learning research since it is closely related to the real-world environment. We study the domain adaption in the sequence labeling problem where the model trained on the source domain data is given. We propose two methods: Self-Adapter and Selective Classifier Training. Self-Adapter is a training method that uses sentence-level pseudo-labels filtered by the self-entropy threshold to provide supervision to the whole model. Selective Classifier Training uses token-level pseudo-labels and supervises only the classification layer of the model. The proposed methods are evaluated on data provided by SemEval-2021 task 10 and Self-Adapter achieves 2nd rank performance.

1 Introduction

Domain adaptation (DA) is the task of applying an algorithm trained on a source domain data to a different target domain data with limited/undefined labels. DA has gotten significant attention as an alternative of fine-tuning approach (Ganin and Lempitsky, 2015; Saito et al., 2018; Tzeng et al., 2017), especially in situations rich supervision is not possible (Morerio et al., 2018). DA is an important way of overcoming the data shortage of deep learning since it enables the utilization of knowledge from other labeled data.

Source-free DA is then proposed to cope with such data sharing in the general setting of DA, the data distribution in the source domain and the target domain are related but different (Storkey and Sugiyama, 2007), and annotated samples from the source domain are available during the training process. However, many of the data resources are not allowed to be shared in real-life environments as there are increasing concerns for privacy issues. For example, Twitter has a regulation that prevents sharing tweet text. The policy is even more rigorous in the financial/clinical domain under the privacy protection issue.

Unlike conventional DA, one can not get access to the source domain data in source-free DA but is provided a model trained on the source domain data. About source-free DA in computer vision, several approaches have been proposed; (Sahoo et al., 2020) assumes the target domain data is a transformation from the source domain data along natural axes such as brightness and contrast; (Kundu et al., 2020) proposes universal DA that is trained via two-stage learning of procurement and deployment; (Kim et al., 2020) progressively updates the target model with pseudo-labels which are selected under self-entropy criterion.

As for natural language processing (NLP), the application of source-free DA is slightly more complicated since sentences are usually considered as having discrete representations. In this context, SemEval-2021 task 10 has proposed a challenge that is related to source-free domain adaptation for semantic processing.

In this paper, we propose **Self-Adapter** for the *time expression recognition* sub-task in SemEval-2021 task 10. Following (Kim et al., 2020), we employ pseudo-labels from the target domain to further supervise the model trained on the source domain data, while the entropy-based evaluation of reliable pseudo-labels is adopted in consideration of the discrete text data. In addition, we adopt *Sloughing* trick to prevent over-fitting.

To demonstrate the efficacy of the Self-Adapter, we evaluate the proposed method on the dataset by (Laparra et al., 2018). We also compare the proposed method with several variations and another method we come up with, named **Selective Classifier Training (SCT)**. In the end, the Self-Adapter has achieved 2nd rank in the official evaluation period getting 0.811 F_1 which is 1.7 percentage

Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021), pages 452–457
Bangkok, Thailand (online), August 5–6, 2021.

points higher than the RoBERTa-based sequence tagging model pre-trained only on source data.

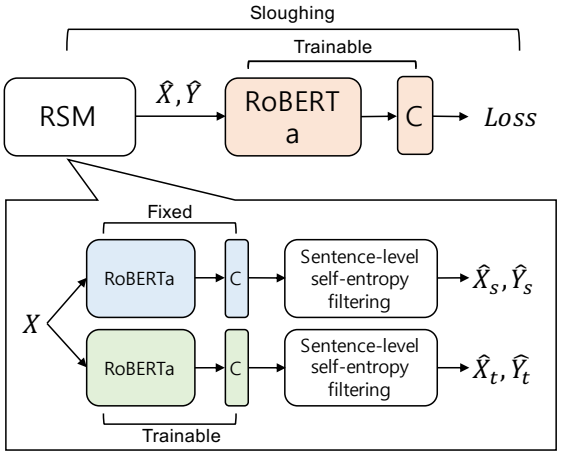

Figure 1: Training pipeline of Self-Adapter. At the beginning of every stage of training, RSM is initialized with 'reliable' samples generated by both fixed and trainable models. A trainable model is supervised using samples - *source-oriented pseudo-labels* and *target-oriented pseudo-labels* - stored in RSM.

2 Systems Description

Our proposed methods have three operations in common: (1) generating pseudo-labels, (2) filtering out pairs of reliable samples and pseudo-labels based on models' self-confidence, and (3) doing supervised learning using the pseudo-labels. We concentrate on sorting out 'reliable' pseudo-labels since training with incorrect labels harms the performance of the model.

Self-entropy is usually treated as an indicator of self-confidence (Zou et al., 2018; Saporta et al., 2020). We adopt normalized self-entropy as the evaluation metric for pseudo-labels:

$$H(x_t) = -\frac{1}{logN_c} \sum l(x_t) log(l(x_t)) \quad (1)$$

where x_t denotes each token that makes up a sentence $X \in \mathbf{X}$. $l(x_t)$ denotes the predicted probability of the predicted label by the classifier, and N_c refers to the total number of labels.

Specifically, we propose two adaptation methods to efficiently fit the model trained on a source domain to a target domain: Self-Adapter and SCT.

2.1 Method 1: Self-Adapter

We propose Self-Adapter which is a self-learning method under the supervision of reliable sample memory (RSM). RSM is a set of data with pseudo-labels that consists of two parts, *source-oriented pseudo-labels* and *target-oriented pseudo-labels*, and each of them represents the knowledge learned from the source domain and new features to learn from the target domain. We further apply a trick called '*Sloughing*' which helps prevent over-fitting. The overall workflow of Self-Adapter is shown in Figure 1.

2.1.1 Reliable Sample Memory

RSM is the pairs of input sentences and the corresponding pseudo-labels obtained from a Siamese-like network structure. Two RoBERTa-based (Liu et al., 2019) classifiers are initialized with a RoBERTa-based sequence tagging model fine-tuned only on source train data, which is given as a baseline model in the task. One of the branch maintains fixed weight parameters while another is fine-tuned during training.

Both branches of the network take a target domain sentence X as an input and output a set of probabilities for labels each token should be assigned to. We utilize the self-entropy as the evaluation metric for the self-confidence of each token. If the self-confidence of each token is smaller than the predefined threshold, the pair of input sentences with the pseudo-labels generated by the model is kept as a part of RSM.

The fixed part of the network consistently outputs the same pairs $(\hat{X}_s, \hat{Y}_s)$ which are called *source-oriented pseudo-labels*. The trainable part of the network outputs different pairs $(\hat{X}_t, \hat{Y}_t)$ called *target-oriented pseudo-labels* after each update and both are stored in the RSM. All sentence-label pairs in RSM, both *source-oriented pseudo-labels* and *target-oriented pseudo-labels*, are used to train the trainable part of the network in a supervised manner. We call the cycle in which RSM are updated as a stage and each stage is composed of several epochs.

2.1.2 *Sloughing* trick

After sufficient update of RSM, we generate pseudo-labels with RSM and do another self-entropy filtering to gain new reliable samples. Subsequently, we re-initialize the trainable part of the network with the parameter of the baseline and train it under the supervision of the new reliable samples. We call this procedure *Sloughing*. Since many of the reliable samples in each RSM update overlaps, over-fitting tends to happen over time.

The *Sloughing* then efficiently prevents over-fitting by newly initializing a model which is not fitted to test data yet.

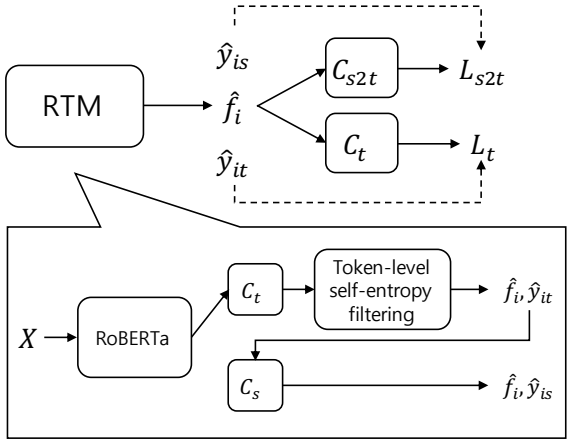

Figure 2: Training pipeline of Selective Classifier Training. RTM is updated at the start of each step of training. In multi-branch network whose two branches C_{s2t} and C_t share a fixed RoBERTa-based feature extractor, loss of C_{s2t} branch is calculated by supervision with *source-oriented token-wise pseudo-labels* and loss of C_t branch is calculated by supervision with *target-oriented token-wise pseudo-labels*.

2.2 Method 2: Selective Classifier Training

Selective Classifier Training is a training method that consists of a RoBERTa-based feature extractor and multi-branch classifiers. The feature extractor and classifiers are initialized with the RoBERTa-based sequence tagging model fine-tuned only on source train data, given as baseline model in the development phase. In SCT, only the classifiers are updated under the supervision of Reliable Token Memory (RTM).

2.2.1 Reliable Token Memory

RTM is the pairs of tokens and their pseudo-labels obtained from a network with two separate branches. Both the branches share the fixed feature extractor which is the same with the feature extractor of the SCT training pipeline. Two classifiers, a trainable classifier C_t and a fixed classifier C_s, make predictions on contextual embedding passed from the feature extractor.

To update RTM, we first get contextual embeddings for all tokens in target domain sentences by putting in all sentences as input of shared feature extractor and we get prediction on each token embeddings. Token embeddings $\hat{f}_i$ whose normalized self-entropy predicted by C_t are lower than the threshold θ are called *reliable token-wise samples*. The pseudo-labels of reliable token-wise samples predicted by C_s are called *source-oriented token-wise pseudo-labels*. The pseudo-labels of reliable token-wise samples predicted by C_t are called *target-oriented token-wise pseudo-labels*. The pairs of *reliable token-wise samples* and their *source-oriented token-wise pseudo-labels*, and the pairs of *reliable token-wise samples* and their *target-oriented token-wise pseudo-labels* consists RTM.

2.2.2 Multi-branch network

With RTM, we train a multi-branch network in which each branch shares a fixed feature extractor. They divide into two classifiers C_{s2t} and C_t. Loss of C_{s2t} branch is calculated by supervision with *source-oriented token-wise pseudo-labels* and loss of C_t branch is calculated by supervision with *target-oriented token-wise pseudo-labels*. RTM updates at the start of each step of training.

The loss function is formulated as

$$L_{total} = (1 - \alpha)L_{s2t} + \alpha L_t \quad (2)$$

where L_{s2t} and L_t indicates loss function of C_{s2t} branch and C_t respectively. α is a weight between two branches. We gradually increase α from 0 to 1 to deal with high instability in the early stages of learning, in the same way as (Kim et al., 2020). In the test phase, we use the classification probability of the C_t branch.

3 Experiments

We evaluate our two models: Self-Adapter, SCT and their variations. The baseline on the development data is a RoBERTa-based sequence tagging model pre-trained on only the source data: de-identified clinical notes from the Mayo Clinic, called Source-Trained. Also, there is another baseline Dev-Tuned on the test data which is the source pre-trained model (i.e., Source-Trained) fine-tuned on the labeled development data. The development data is the annotated news portion of the SemEval-2018 Task 6 data. Test data is a set of annotated documents extracted from food security warning systems. development data consists of 1580 sentences and test data consists of 3911 sentences. The total number of labels is 65, where label 0 indicates non-time entity, and label 1-64 indicates different types of time entities.

Method	F_1	$Precision$	$Recall$
SCT	0.784	0.814	0.756
SA	0.808	0.819	0.797
SA+*Sloughing**	**0.812**	0.822	0.801
SA-filtering	0.771	0.774	0.768
Source-Trained	0.771	0.768	0.775

Table 1: F_1, $Precision$, $Recall$ on development data. The model submitted to the competition is marked with *. SA indicates Self-Adapter and SA+*Sloughing* is a system where *Sloughing* is applied on a model trained with Self-Adapter. SA-filtering is a system whose training pipeline is the same as Self-Adapter except that confidence filtering is not done. Source-Trained is a RoBERTa-based sequence tagging model pre-trained on only the source data: de-identified clinical notes from the Mayo Clinic, given as baseline model in the development phase of the competition.

Method	F_1	$Precision$	$Recall$
SA	0.81	0.874	0.754
SA+*Sloughing**	**0.811**	0.873	0.757
Source-Trained	0.794	0.849	0.746
Dev-Tuned	0.804	0.827	0.782

Table 2: F_1, $Precision$, $Recall$ on the test data. The model submitted to the competition is marked with *. SA indicates Self-Adapter and SA+*Sloughing* is a system where *Sloughing* is applied on a model trained with Self-Adapter. Source-Trained is a RoBERTa-based sequence tagging model pre-trained on only the source data: de-identified clinical notes from the Mayo Clinic and Dev-Tuned is a the source pre-trained model (i.e., Source-Trained) fine-tuned on the labeled development data.

3.1 Experimental setup

For all of our models, we set normalized self-entropy threshold $\theta = 0.1$ except when applying *Sloughing* trick, on which $\theta = 0.01$. We train Self-Adapter for 3 stages. Each iteration consists of 4 epochs with batch size 1 (sentence-level) and the learning rate is fixed as 5e-5. In Self-Adapter, pseudo-labels are updated at every stage. In Self-Adapter combined with *Sloughing*, we apply *Sloughing* for 3 times, 4 epochs training with batch size 1 (sentence-level) is done every time. The learning rate is fixed as 5e-5. In SCT, pseudo-labels are updated every epoch. We train 2 epochs with batch size 4 (token-level) and the learning rate is scheduled with inverse decay scheduler same as (Kim et al., 2020), with initial learning rate 5e-5. We use Adam optimizer in all models.

3.2 Experimental results and analysis

Table 1 and Table 2 shows the performance of the proposed methods on development and test data respectively. Each method is evaluated with $Precision$, $Recall$, and F_1. Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. $Recall$ is the ratio of correctly predicted positive observations to all observations in the actual class. F_1 is the weighted average of Precision and Recall. Our major concern is F_1, which is the most preferred indicator of accuracy in text classification tasks.

On both data, Self-Adapter combined with *Sloughing* performs the best in F_1 and Self-Adapter performs the second-best. SCT does not provide significant improvement of F_1 compared to Self-Adapter. Self-Adapter without confidence filtering performs almost the same as Source-Trained on every evaluation metric.

3.2.1 Impact of confidence filtering

Our confidence filtering proves to be effective in dealing with the uncertainty of pseudo-labels. Self-Adapter, whose core is confidence filtering, increases 3.7, 1.6 percentage points of F_1 on development data and test data for each. The system whose training pipeline is the same as Self-Adapter except that confidence filtering is not done performs almost the same as the Source-Trained.

3.2.2 Necessity of training feature extractor

Well-trained BERT embeddings contain both syntactic (Hewitt and Manning, 2019) and semantic (Coenen et al., 2019) information of words. However, this is only when the model is fine-tuned with data from the domain same as the target domain. It is well known that embedding models trained on different domains poorly capture the domain-specific vocabularies and word semantics due to domain shift. (Sarma et al., 2018)

Since RoBERTa is a BERT-based language model, the same issue arises on RoBERTa used in this task. Thus if the feature extractor used for embedding words is fixed during training, the embeddings obtained do not provide sufficient information to the classifier, resulting in a limitation to improving performance. This is also shown through experimental results in which Self-Adapter outperforms SCT.

3.2.3 Inefficiency of *Sloughing*

In Self-Adapter, the model learns from almost all sentences in development data and test data. Only 333 sentences out of 1580 and 917 sentences out of 3911 were filtered in development data and test data for each despite the high threshold we set (θ = 0.1). It affects the magnitude of the effect of the *Sloughing* in our method. *Sloughing* improves performance on both development and test data, but not enough to be taken as meaningful. 0.04 percentage points of F_1 on development data and 0.1 percentage points of F_1 on test data increase by application of *Sloughing*.

Somewhat discouraging effect of *Sloughing* is due to the setting of our task, in which training is done with almost all samples in test data, despite confidence filtering. We expect *Sloughing* to be more effective in the setting where the bigger proportion of samples are filtered and thus the ability for generalization on unseen data is more important. However, verification of these hypotheses will be carried out as a follow-up study.

4 Conclusion

In this paper, we propose novel training methods Self-Adapter and Selective Classifier Training that improve model performance on the target domain only by leveraging the RoBERTa-based model pretrained on source data. Both models rely on self-learning with highly credible pseudo-labels that are filtered based on self-entropy, differ only in the range of trainable parts. Also, we propose *Sloughing* trick to prevent over-confidence of the model by softening the network output. Our work is highly applicable in the real world since we have achieved remarkable improvement in performance using only a few test data which is not annotated at all, without any manual supervision.

Acknowledgments

This work was supported by the Technology Innovation Program (10073144, Developing machine intelligence based conversation system that detects situations and responds to human emotions) funded By the Ministry of Trade, Industry & Energy(MOTIE, Korea)

References

Andy Coenen, Emily Reif, Ann Yuan, Been Kim, Adam Pearce, Fernanda Viégas, and Martin Wattenberg. 2019. Visualizing and Measuring the Geometry of BERT. In *proceedings of the 33rd Conference on Neural Information Processing Systems*.

Yaroslav Ganin and Victor Lempitsky. 2015. Unsupervised Domain Adaptation by Backpropagation. In *International Conference on Machine Learning*, pages 1180–1189.

John Hewitt and D Christopher Manning. 2019. A Structural Probe for Finding Syntax in Word Representations. In *proceedings of Association for Computational Linguistics*.

Youngeun Kim, Sungeun Hong, Donghyeon Cho, Hyoungseob Park, and Priyadarshini Panda. 2020. Domain Adaptation without Source Data. *arXiv preprint arXiv:2007.01524*.

Jogendra Nath Kundu, Naveen Venkat, R Venkatesh Babu, et al. 2020. Universal Source-Free Domain Adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4544–4553.

Egoitz Laparra, Dongfang Xu, Ahmed Elsayed, Steven Bethard, and Martha Palmer. 2018. Semeval 2018 task 6: Parsing Time Normalizations. In *proceedings of the 12th International Workshop on Semantic Evaluation*, pages 88–96.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A Robustly Optimized BERT Pretraining Approach. *arXiv preprint arXiv:1907.11692*.

Pietro Morerio, Jacopo Cavazza, and Vittorio Murino. 2018. Minimal-Entropy Correlation Alignment for Unsupervised Deep Domain Adaptation. In *proceedings of International Conference on Learning Representations*.

Roshni Sahoo, Divya Shanmugam, and John Guttag. 2020. Unsupervised Domain Adaptation in the Absence of Source Data. *arXiv preprint arXiv:2007.10233*.

Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. 2018. Maximum Classifier Discrepancy for Unsupervised Domain Adaptation. In *Proceedings of the IEEE Con- ference on Computer Vision and Pattern Recognition*, pages 3723–3732.

Antoine Saporta, Tuan-Hung Vu, Matthieu Cord, and Patrick Pérez. 2020. Esl: Entropy-guided Self-supervised Learning for Domain Adaptation in Semantic Segmentation. In *proceedings of Computer Vision and Pattern Recognition*.

Prathusha Sarma, K, Yingyu Liang, and William Sethares, A. 2018. Domain Adapted Word Embeddings for Improved Sentiment Classification. In *Proceedings of the Workshop on Deep Learning Approaches for Low-Resource NLP*.

Amos Storkey and Masashi Sugiyama. 2007. Mixture Regression for Covariate Shift. In *proceedings of Neural Information Processing Systems*, pages 1337–1344.

Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. 2017. Adversarial Discriminative Domain Adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, page 7167–7176.

Yang Zou, Zhiding Yu, Xiaofeng Liu, Vijayakumar Bhagavatula, and Jinsong Wang. 2018. Confidence Regularized Self-Training. In *proceedings of Computer Vision and Pattern Recognition*.