Pseudo-Label Guided Unsupervised Domain Adaptation of Contextual Embeddings

Tianyu Chen[1], Shaohan Huang[2], Furu Wei[2], Jianxin Li[1]*
Beihang University[1], Microsoft Research Asia[2]
tianyuc@buaa.edu.cn shaohanh@microsoft.com
fuwei@microsoft.com lijx@buaa.edu.cn

Abstract

Contextual embedding models such as BERT can be easily fine-tuned on labeled samples to create a state-of-the-art model for many downstream tasks. However, the fine-tuned BERT model suffers considerably from unlabeled data when applied to a different domain. In unsupervised domain adaptation, we aim to train a model that works well on a target domain when provided with labeled source samples and unlabeled target samples. In this paper, we propose a pseudo-label guided method for unsupervised domain adaptation. Two models are fine-tuned on labeled source samples as pseudo labeling models. To learn representations for the target domain, one of those models is adapted by masked language modeling from the target domain. Then those models are used to assign pseudo-labels to target samples. We train the final model with those samples. We evaluate our method on named entity segmentation and sentiment analysis tasks. These experiments show that our approach outperforms baseline methods.

1 Introduction

Contextualized embeddings have become the foundations of many state-of-the-art natural language processing technologies (Devlin et al., 2018; Han and Eisenstein, 2019; Straková et al., 2019). Pre-trained contextualized embeddings can be used for many downstream tasks and be incorporated into an end-to-end system, allowing the embeddings to be fine-tuned from task-specific labeled data (Akbik et al., 2019a,b, 2018; Beltagy et al., 2019).

One of the problems with contextual embedding models is that although fine-tuned models perform well on the samples generated from the same distribution as the training samples, they suffer considerably from unlabeled data when applied to a different domain (Saito et al., 2017; Rietzler et al., 2019; Ruder and Plank, 2018). For example, a named entity segmentation model trained on a news dataset fails to predict correctly on social media data such as Twitter. Because collecting many labeled samples in various domains is expensive, it is important to adapt contextual embedding model to different domains in unsupervised setting.

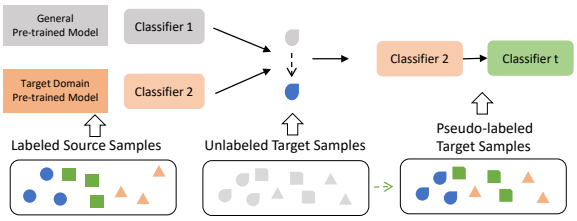

Figure 1: Overview of our training framework. We jointly fine-tune general pre-trained model and target domain pre-trained model with labeled source data. Then we generate pseudo-labels on target samples. Finally, we train the final model with the pseudo-labeled samples.

Many domain adaptation methods of neural networks in NLP have been proposed in the past several years (Li, 2012; Ziser and Reichart, 2019, 2018; Cui et al., 2018; Louizos et al., 2015; Ganin et al., 2015; Mou et al., 2016). Our work focuses on unsupervised domain adaptation of contextual embeddings. We aim to fine-tune a pre-trained model that works well on a target domain when provided with labeled source samples and unlabeled target samples. With no access to the labels in target domain data, it is very difficult to adapt when the divergence of the label distribution between the source domain and the target domain is huge.

Some of current methods propose a simple unsupervised domain-adaptive method, using a masked language modeling objective over unlabeled text in the target domain (Rochette et al., 2019; Han and Eisenstein, 2019; Gururangan et al., 2020). They first learn discriminative representations for the tar-

*Corresponding author.

Proceedings of the Second Workshop on Domain Adaptation for NLP, pages 9–15
April 20, 2021. ©2021 Association for Computational Linguistics

get domain and then fine-tune a domain-adapted model with labeled source samples. Although self-supervised fine-tuning in the target domain improves the generalization of the pre-trained model for the target domain, an adapted model cannot capture the task-specific pattern of the target domain only using labeled source samples to fine-tune. We expect the adapted model to not only acquire some target-discriminative language representations but also to obtain some task-specific features of target domain.

In this paper, we propose a pseudo-label guided method for unsupervised domain adaptation. As shown in Figure 1, two models are jointly fine-tuned on labeled source samples as pseudo labeling models. We design a multiview constraint loss to encourage those two model to make predictions based on different view-point. To learn representations for the target domain, one of those models is adapted by masked language modeling from the target domain. Then those models are used to assign pseudo-labels to target samples. Pseudo-labeled target samples will provide target discriminative information to the model. We train the final model with the pseudo-labeled samples.

We evaluate our method both on named entity segmentation (NES) and sentiment analysis (SA) tasks. We find that our pseudo-label guided method outperforms baseline methods. Moreover, we demonstrate the multiview constraint significantly improves performance of our method.

2 Methodology

2.1 Overview

In unsupervised domain adaptation, we aim to train a model that works well on a target domain when provided with labeled source samples and unlabeled target samples.

As illustrated in Figure 1, **P**seudo-label **G**uided unsupervised **A**daptation (**PGA**) consists of three steps: jointly fine-tuning, pseudo label generation and final adaptation. First, we initialize two pre-trained models with the same architecture as the general pre-trained model and the target domain pre-trained model (TDPM) which leverages language modeling objective on unlabeled data from target domain. In the second step, we use those two fine-tuned model to make predictions for unlabeled target samples. If both models agree with the prediction and two prediction scores exceed a threshold, the prediction is regarded as a pseudo label. Finally, all pseudo labels are collected to fine-tune the target domain pre-trained model and complete the adaptation.

2.2 Jointly Fine-tuning

In the first stage, we jointly fine-tune the general pre-trained model and the target domain pre-trained model on the source domain data to obtain two classification models F_1 and F_2. Their predictions are utilized to give pseudo-labels. We assume each sample in the source domain can be denoted as (X_i, Y_i), where X is a text sequence and Y is a label sequence for NES task or a label for SA task.

For named entity segmentation task, the token representations are fed into an output layer at the output. For sentiment analysis, the [CLS] representation is fed into an output layer for classification (Devlin et al., 2018).

Inspired by the asymmetric tri-training adaptation method (Saito et al., 2017), we design a multiview constraint loss to encourage model F_1 and F_2 to make predictions based on different viewpoints. We add the term $|W_1^T W_2|$ to the cost function, where W_1, W_2 denote output layers weights of model F_1 and F_2. With this constraint, each model learns from different features. The objective of learning F_1, F_2 is defined as:

$$\begin{aligned} E(\theta_{F_1}, \theta_{F_2}) = CE_{loss}(F_1(x_i), y_i) + \\ CE_{loss}(F_2(x_i), y_i) + \lambda |W_1^T W_2| \end{aligned} \quad (1)$$

where CE_{loss} denotes the standard cross entropy loss and we decide the trade-off parameter λ based on validation split. With the multiview learning objective, the pseudo labels can be more informative and improve model accuracy.

2.3 Pseudo Label Generation

After jointly fine-tuning, classification model F_1 and F_2 are used to generate pseudo labels. Pseudo labels will provide target-discriminative information to the model. However, since they certainly contain false labels, we have to pick up reliable pseudo-labels.

For text sequence X in the unlabeled target domain data, we add pseudo annotations Y to the sequence in the NES task and add single pseudo label Y to the sequence in the SA task.

There are two requirements for pseudo label assignment. Take the NES task as an example. First, for each token X_i in the text sequence, when C_i^1 and C_i^2 denote the class which have the maximum

predicted probability for X_i from model F_1 and model F_2 respectively, we require $C_i^1 = C_i^2$, which means the two models agree with the prediction. The second requirement is that the probability of C_i^1 or C_i^2 exceeds the threshold parameter, which we set as 0.5 in the experiment. We suppose that unless two models are confident of their predictions, the prediction is not reliable. If the two requirements are satisfied, the label is added to the pseudo target samples.

Intuitively, if the two domains are closely related, the pseudo labels are assigned to a large portion of target samples while distant domains will reduce the amount of agreement. We expect the threshold of agreement to keep the pseudo labels reliable and maintain a probable number of samples.

2.4 Final Adaptation

We use the pseudo-labeled target samples to construct a training set of the target domain and further fine-tune the target domain pre-trained model with this training set. Since the accuracy of pseudo labels can not be assured, we use a smaller learning rate and fewer training steps to fine-tune F_2 with pseudo labels.

The whole training algorithm is depicted in Algorithm 1, where we take labeled source samples and unlabeled target samples and output the adapted model.

Algorithm 1 We jointly fine-tune two models and generate pseudo labels for final adaptation.

Input:
$S = \{(x_i, t_i)\}_{i=1}^m, T = (x_j)_{j=1}^n$
Train TDPM on T with language modeling objective
Initialize F_1 with original BERT
Initialize F_2 with TDPM
Train F_1, F_2 with Equation 1
Initialize $T_l = \varnothing$
for x_j **in** T **do**
  $y_j^1 = F_1(x_j)$
  $y_j^2 = F_2(x_j)$
  $C_j^1 = argmax(y_j^1)$
  $C_j^2 = argmax(y_j^2)$
  if $C_j^1 == C_j^2$ and $max(y_j^1, y_j^2) > threshold$ **then**
    Add (x_j, y_j^1) to T_l
  end if
end for
Train F_2 on T_l with supervised learning
Output:F_2

3 Experiments

3.1 Datasets

Named Entity Segmentation (NES) The named entity segmentation is a typical task of the sequence labeling task. Different from named entity recognition, we only predict the "BIO" format in a sequence, which divides the sequence into entity chunks without deciding which class the entity chunk belongs to. We choose the shared task of the 2016 workshop on Noisy User Text (WNUT; Strauss et al., 2016) as target domain and the canonical CoNLL 2003 shared task (Tjong Kim Sang and De Meulder, 2003) as the source domain. In the WNUT task, the corpus was from Twitter, which is an open-domain source and the data of the CoNLL 2003 task was annotated on a corpus of newstext.

Sentiment Analysis (SA) The sentiment analysis is a sequence classification task. Since we need to assign a sentiment class to each whole sentence, the task is more relied on contextualized level information. We choose 3 domains from open Amazon review data (He and McAuley, 2016) including books, electronics and kitchens, following the settings of Ruder and Plank's domain adaptation survey (Ruder and Plank, 2018). The data statistics and hyper-parameters see Appendix A and B. The source code* and sentiment analysis data set will be released at a future date.

3.2 Baselines

We evaluate the following systems:

Source only: This baseline directly fine-tunes the pre-trained BERT on the source domain.

Frozen BERT: This baseline first learns from unlabeled data from target domain by language modeling. Then it freezes the BERT encoder and only optimizes the classifier layer.

AdaptaBERT: This baseline first learns from unlabeled data from target domain by language modeling. Then it fine-tunes target domain pre-trained model with labeled source samples†.(Han and Eisenstein, 2019).

PGA: Our pseudo-label guided unsupervised adaptation method in Section 2.

*Implementation retrieved from https://github.com/huggingface/transformers

†Use the released code to conduct our experiment https://github.com/xhan77/AdaptaBERT

Model	WNUT (target)	CoNLL (source)
Source only	56.52	97.69
AdaptaBERT	63.81	97.67
PGA w/o MC	64.12	96.89
PGA	**64.32**	96.82
Upper Bound	65.81	82.64

Table 1: Named entity segmentation performance on the WNUT test set (target) and CoNLL test set A (source). The F1 score of our method is very close to the upper bound of the **target domain**.

Model	Books ↓ Electronics	Electronics ↓ Books	Books ↓ Kitchens	Kitchens ↓ Books
Source only	45.68	47.84	45.28	49.56
Frozen BERT	30.52	31.76	30.16	31.44
AdaptaBERT	46.56	49.32	47.76	51.32
PGA w/o MC	45.33	50.61	48.60	53.18
PGA	**50.81**	**50.62**	**49.28**	**53.74**
Upper Bound	60.04	57.44	59.68	57.44

Table 2: Multi-domain sentiment analysis adaptation. All results are evaluated with accuracy score.

PGA w/o MC: Our pseudo-label guided unsupervised adaptation method. Notice that the multiview constraint (MC) is not used.

Upper Bound: In *supervised learning*, we finetune the target domain pre-trained model directly on target training set and evaluate.

3.3 Results

As indicated in Table 1, AdaptaBERT shows strong performance of unsupervised adaptation, achieving a much better F1 score than zero-shot setting in source only. Moreover, even without multiview constraint, our PGA method performs better than AdaptaBERT. With multiview constraint, our method learns a higher quality of pseudo labels, which pushes the F1 score closer to the upper bound.

We present the domain adaptation results of sentiment analysis at Table 2, without a multiview constraint, the quality of pseudo labels can not be assured and sometimes leads to a drop in performance for target domain. However, our PGA method with multiview constraint can improve model accuracy in most scenarios. We can observe that supervised learning still greatly outperforms unsupervised methods. There is more room to improve for unsupervised domain adaptation in sentiment analysis task.

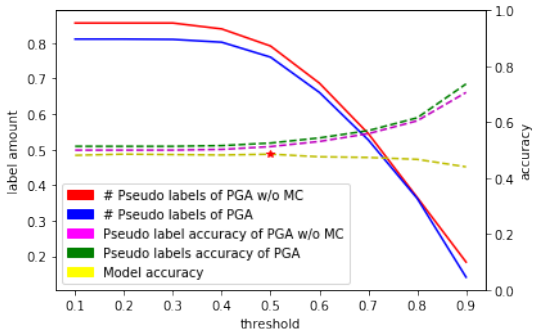

Figure 2: The threshold influences the amount of pseudo labels. We evaluate on books to electronics domain adaptation of sentiment analysis.

Domain Adaptation	PGA	PGA w/o MC
Books → Electronics	52.97	51.25
Books → Kitchens	51.94	51.03
Electronics → Books	56.72	55.34
Kitchens → Books	59.26	58.32

Table 3: The accuracy of pseudo labels.

3.4 Pseudo Labels Quality

We scientifically evaluate the quality of pseudo labels generated by our method. First, we find the amount of pseudo labels is closely related to the value of our threshold in Figure 2. When the threshold is set above 0.5, the number of pseudo labels drops quickly while the accuracy of pseudo labels increases a lot. With multiview constraint, our PGA method tend to generate fewer labels with higher accuracy. It is observed that higher threshold may not benefit the final model accuracy, partly due to the significant drop of the number of pseudo labels.

We also evaluate the accuracy of pseudo labels in the SA task. The results are illustrated in Table 3. We find in different domain adaptation settings, the multiview constraint can improve the quality of pseudo labels.

4 Conclusion

We propose a new unsupervised domain adaptation method guided by pseudo labels. Generated by general pre-trained model and target domain pre-trained model with multiview constraint, the pseudo labels of unlabeled target data are more reliable and benefit model performance on the target domain. Experiments show that our approach achieves very promising results on different NLP downstream tasks.

References

Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. 2019a. Flair: An easy-to-use framework for state-of-the-art nlp. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59.

Alan Akbik, Tanja Bergmann, and Roland Vollgraf. 2019b. Pooled contextualized embeddings for named entity recognition. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 724–728, Minneapolis, Minnesota. Association for Computational Linguistics.

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1638–1649, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. Scibert: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3606–3611.

Xia Cui, Noor Al-Bazzas, DANUSHKA Bollegala, and FP Coenen. 2018. A comparative study of pivot selection strategies for unsupervised domain adaptation. *The Knowledge Engineering Review*.

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

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2015. Domain-adversarial training of neural networks.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of ACL*.

Xiaochuang Han and Jacob Eisenstein. 2019. Unsupervised domain adaptation of contextualized embeddings for sequence labeling. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4229–4239.

Ruining He and Julian McAuley. 2016. 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*, pages 507–517.

Qi Li. 2012. Literature survey: domain adaptation algorithms for natural language processing. *Department of Computer Science The Graduate Center, The City University of New York*, pages 8–10.

Christos Louizos, Kevin Swersky, Yujia Li, Max Welling, and Richard Zemel. 2015. The variational fair autoencoder.

Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. 2015. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 43–52.

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

Alexander Rietzler, Sebastian Stabinger, Paul Opitz, and Stefan Engl. 2019. Adapt or get left behind: Domain adaptation through bert language model finetuning for aspect-target sentiment classification. *arXiv preprint arXiv:1908.11860*.

Alexandre Rochette, Yadollah Yaghoobzadeh, and Timothy J Hazen. 2019. Unsupervised domain adaptation of contextual embeddings for low-resource duplicate question detection. *arXiv preprint arXiv:1911.02645*.

Sebastian Ruder and Barbara Plank. 2018. Strong baselines for neural semi-supervised learning under domain shift. *CoRR*, abs/1804.09530.

Kuniaki Saito, Yoshitaka Ushiku, and Tatsuya Harada. 2017. Asymmetric tri-training for unsupervised domain adaptation. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2988–2997. JMLR. org.

Jana Straková, Milan Straka, and Jan Hajic. 2019. Neural architectures for nested NER through linearization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5326–5331, Florence, Italy. Association for Computational Linguistics.

Benjamin Strauss, Bethany Toma, Alan Ritter, Marie-Catherine de Marneffe, and Wei Xu. 2016. Results of the WNUT16 named entity recognition shared task. In *Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT)*, pages 138–144, Osaka, Japan. The COLING 2016 Organizing Committee.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Yftah Ziser and Roi Reichart. 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, New Orleans, Louisiana. Association for Computational Linguistics.

Yftah Ziser and Roi Reichart. 2019. Task refinement learning for improved accuracy and stability of unsupervised domain adaptation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5895–5906.

A Hyperparameters

We use a bert-base-cased model from huggingface [‡] as initial parameter checkpoint. In supervised learning stage, we fine-tune the model for 3 to 5 epochs with batch size between [16, 32] and learning rate between [2e-5, 5e-5]. The weight decay of model parameters has been set as 0.1. Adam optimizer has been adopted with a warm up ratio of 0.1. In the unsupervised language modeling stage, the rate of masking tokens has been set as 0.15.We train the model for 3 epochs and adopt the same optimizer settings in supervised learning. We set the threshold parameter as 0.5 for the best performance model in both NES and SA task.

B Data Statistics

In this section, we will introduce the data statistics of our named entity segmentation and sentiment analysis . In Table 4, the data are from origin AdaptaBERT (Han and Eisenstein, 2019), of which Twitter dataset has more unlabeled dev data than labeled train data. In Table 5, the data is collected from open Amazon review (He and McAuley, 2016; McAuley et al., 2015). We process the data into 3 domains of balanced datasets. In the pre-processing stage, we exclude the text whose length is too short (shorter than 1 valid token) or too long (more than 256 valid tokens). The sentiment analysis data has been shared via google drive [§].

Dataset	Train	Dev
CoNLL	14986	10,000
Twitter	2394	3852

Table 4: Data statistics of named entity segmentation.

Dataset	Train	Dev
Books	100,000	10,000
Electronics	100,000	10,000
Kitchens	100,000	10,000

Table 5: Data statistics of sentiment analysis. The train data and dev data of each domain is balanced, which means the number of samples of each class is the same.

[‡] https://github.com/huggingface/transformers

[§] https://drive.google.com/drive/folders/1fdmD08pZUzN3WbgxEu_Hv9lq4ZKmq7J3?usp=sharing

C Experiment details

We use a single Tesla P40 card to conduct all our experiments. The average runtime of each approach is 3 minutes for named entity segmentation and 30 minutes for sentiment analysis. The number of parameters in our model is about 110M , same as the bert-base model.