Leveraging Training Dynamics and Self-Training for Text Classification

Tiberiu Sosea **Cornelia Caragea**
Computer Science
University of Illinois Chicago
tsosea2@uic.edu cornelia@uic.edu

Abstract

The effectiveness of pre-trained language models in downstream tasks is highly dependent on the amount of labeled data available for training. Semi-supervised learning (SSL) is a promising technique that has seen wide attention recently due to its effectiveness in improving deep learning models when training data is scarce. Common approaches employ a teacher-student self-training framework, where a teacher network generates pseudo-labels for unlabeled data, which are then used to iteratively train a student network. In this paper, we propose a new self-training approach for text classification that leverages training dynamics of unlabeled data. We evaluate our approach on a wide range of text classification tasks, including emotion detection, sentiment analysis, question classification and gramaticality, which span a variety of domains, e.g, Reddit, Twitter, and online forums. Notably, our method is successful on all benchmarks, obtaining an average increase in F1 score of 3.5% over strong baselines in low resource settings. We make our code avilable at https://github.com/tsosea2/AUM-ST.

1 Introduction

Deep learning models have achieved impressive performance in most supervised learning settings. However, supervised approaches struggle when labeled data is scarce, since they often suffer from overfitting (Xie et al., 2020a). Semi-supervised learning (SSL) is a powerful approach (Miyato et al., 2018; Sajjadi et al., 2016b; Laine and Aila, 2017; Tarvainen and Valpola, 2017; Berthelot et al., 2019b,a; Xie et al., 2020a; Lee et al., 2013; Sajjadi et al., 2016a; Rosenberg et al., 2005; Verma et al., 2021; Miyato et al., 2016; Chen et al., 2020a; Gururangan et al., 2019a; Zhang et al., 2017b; Izmailov et al., 2020; Sachan et al., 2019) that can overcome this drawback by leveraging unlabeled data, which usually comes in much larger quantities and is easier to collect.

A popular category of SSL methods in text classification is self-training (McLachlan, 1975; Xie et al., 2020b; Rasmus et al., 2015; Scudder, 1965; Mukherjee and Hassan Awadallah, 2020), an iterative approach that uses a trained teacher model to produce pseudo-labels for unlabeled examples, then uses these labels to train a student model, followed by repeating the process with the student as a new teacher until a convergence criterion is met. The quality of the pseudo-labels in the teacher-student framework is an important factor in the self-training process. In supervised learning, noisy labels are problematic and can negatively impact the generalization performance (Zhang et al., 2017a) especially for deep neural networks, which can attain zero training error on any dataset (Zhang et al., 2016). This phenomenon applies to self-training as well since the student model predictions are optimized towards potentially noisy pseudo-labels. To address this drawback, popular self-training methods (Xie et al., 2020b) mask out examples that the teacher model is not confident about. However, relying only on the teacher's confidence in predictions can be problematic especially if the teacher model is not well calibrated (Guo et al., 2017) or has poor performance.

In this work, we investigate the impact of the pseudo-label quality over the performance of self-training methods in text classification and show that designing more sophisticated quality assurance measures for the teacher pseudo-labels leads to an improvement in generalization performance. We, hence, propose a novel self-training method that leverages *training dynamics* to assess the adequacy of the teacher pseudo-labels. In a nutshell, instead of using only the teacher's current *beliefs* about an unlabeled example (i.e., the confidence) to decide if an example should be masked or not, our method also analyzes how pseudo-labeled examples *behave* during training. Specifically, we leverage Area Under the Margin (AUM) (Pleiss et al., 2020) from

Findings of the Association for Computational Linguistics: EMNLP 2022, pages 4779 - 4791

supervised learning, which captures the divergence between the annotated label and the predicted label during training. AUM is calculated as the average difference between the logit corresponding to the gold annotated label and the largest *other* logit. Prior work has shown that a low AUM score correlates well with an example being mislabeled.

Our approach, which we call AUM-ST, extends AUM to unlabeled data and provides a more robust and effective mechanism of identifying noisy pseudo-labels compared to the current approaches based only on confidence. For each unlabeled example, AUM-ST computes the average logit difference between the teacher pseudo-label and the largest other logit during training. An unlabeled example with low AUM indicates that there is a constant tension between its assigned (potentially incorrect) pseudo-label and the hidden true class. Our method therefore masks pseudo-labeled examples with low AUM. Critically, unlike the vanilla AUM, where the annotated labels are constant, in AUM-ST the pseudo-labels are variable and dependent on the teacher network. In each self-training iteration, as the self-training process progresses, the teacher starts generating more qualitative pseudo-labels, which are then used to further improve the student. In a way, our AUM-ST can be viewed as a method to enforce a strict learning *curriculum* (Gong et al., 2016; Kervadec et al., 2019; Yu et al., 2020), where challenging unlabeled examples are not used until the teacher network is able to produce adequate pseudo-labels for them.

To show the effectiveness of our approach, we test it on a diverse range of text classification tasks, ranging from emotion detection and sentiment analysis to gramaticality and question classification. Notably, AUM-ST is extremely effective on all benchmarks in low resource settings, obtaining an average improvement in accuracy over a baseline BERT (Devlin et al., 2019) model of 3.5% using 200 examples per class and 8.3% improvement using as few as 20 examples per class.

Our contributions are as follows: **1)** We introduce a new self-training method for text classification that leverages training dynamics to enforce high-quality teacher pseudo-labels during training. **2)** We evaluate our approach and demonstrate its effectiveness on eight text classification tasks in low resource settings. **3)** We perform a comprehensive analysis of our approach and show its effectiveness in removing noisy teacher pseudo-labels during training.

2 Related Work

We first discuss related work on semi-supervised learning for text classification. Second, we zoom in to self-training, a type of SSL that is the core of our AUM-ST. Finally, we discuss approaches of learning with label noise.

Semi-supervised Learning in NLP Semi-supervised learning has attracted much attention in the NLP community (Gururangan et al., 2019b; Yang et al., 2015; Clark et al., 2018; Chen et al., 2020b; Yang et al., 2017; Chen et al., 2020b; Xie et al., 2020a; Mukherjee and Awadallah, 2020b), since unlabeled data is often much easier to acquire compared to labeled data. For example, Miyato et al. (2016) used adversarial perturbations to text in the word embedding space. Yang et al. (2019) used a hierarchy structure to propagate supervision from high-level labels to lower-level labels, while Clark et al. (2018) introduced cross-view training, where a model makes auxiliary predictions only seeing parts of the input text and is trained to match the predictions when given the entire input. Xie et al. (2020a) used data augmentations on unlabeled examples and trained the model to output the same predictions when fed clean or augmented versions of the same input. Mukherjee and Awadallah (2020b) introduced uncertainty estimates into self-training, a particular type of SSL where a teacher and a student model are iteratively trained using labeled and unlabeled data. Self-training is the core of our AUM-ST, hence we detail it further in the next paragraph.

Self-Training Our AUM-ST approach builds upon previous works on self-training (Miyato et al., 2018; Sajjadi et al., 2016b; Laine and Aila, 2017; Tarvainen and Valpola, 2017; Berthelot et al., 2019b,a; Xie et al., 2020a; Lee et al., 2013; Sajjadi et al., 2016a; Rosenberg et al., 2005; Verma et al., 2021; Miyato et al., 2016; Chen et al., 2020a; Gururangan et al., 2019a; Zhang et al., 2017b; Izmailov et al., 2020; Sachan et al., 2019), but replaces the traditional confidence thresholding data filtering mechanism with a more effective approach that takes into account the training dynamics of unlabeled examples.

Self-training is an SSL method where a single model is repeatedly trained on both labeled and unlabeled data, until a convergence criterion is met. The model selects which unlabeled data to train on

using its own predictions. Concretely, traditional self-training follows these steps: **1)** Train a teacher model $\mathcal{M}$ on a labeled set L. **2)** Use $\mathcal{M}$ to make predictions and obtain pseudo-labels on a set of unlabeled examples U. **3)** Optionally, filter out unlabeled examples using a criterion. For example, in traditional self-training, unlabeled examples where the teacher confidence is not high enough are ignored. **4)** Use both the labeled set L and the generated pseudo-labeled set to train a new student model $\mathcal{M}^{'}$. **5)** Continue to step **2** with the student as the new teacher (i.e., $\mathcal{M} \leftarrow \mathcal{M}^{'}$).

Learning with Label Noise Several approaches to achieve label noise robustness have been proposed. For example, Goldberger and Ben-Reuven (2016) proposed adding a noise layer in the neural network architecture, whose parameters can be learned for an accurate correct label estimation. Saxena et al. (2019) introduced a curriculum-learning approach that uses learnable data parameters and ranks the importance of examples in the learning process. These parameters are then leveraged to decide the data to use at different training stages. Liu and Guo (2020) on the other hand proposed to alter the loss function to make it more robust in the face of label noise. To this end, they introduced Peer Loss Functions, which evaluate predictions on both the samples at hand, as well as carefully automatically constructed *peer* samples. Other approaches designed techniques to accurately identify and eliminate potentially mislabeled instances (Brodley and Friedl, 1999; Pleiss et al., 2020). Our work builds on the latter approaches; we leverage Area Under the Margin (Bartlett et al., 2017; Pleiss et al., 2020; Elsayed et al., 2018; Jiang et al., 2018) and adapt it to our self-training setup. We emphasize that most of these methods for learning with label noise can be adapted to our setting, and these are potential future directions for our work.

3 Our method

In this section, we first provide background information on the vanilla AUM (Pleiss et al., 2020) metric. Next, we introduce our proposed AUM-ST and detail the various procedures that we used to improve its performance.

3.1 Background

We start by introducing Area Under the Margin (AUM) (Pleiss et al., 2020), a metric from supervised learning based on training dynamics that can characterize training examples with respect to their contribution to generalization. AUM is defined as the margin averaged across all training epochs $\mathcal{T}$. Specifically, at an arbitrary epoch $t \in \mathcal{T}$, the margin is:

$$M^t(x, y) = z_y - max_{y!=i}(z_i) \quad (1)$$

where $M^t(x, y)$ is the margin of example x with true label y, z_y is the logit corresponding to the true label y, and $max_{y!=i}(z_i)$ is the largest *other* logit corresponding to label i not equal to y. Intuitively, the margin measures how different a true label is compared to a model's *belief* at each epoch t. Therefore, the AUM of x is defined as:

$$AUM(x, y) = \frac{1}{\mathcal{T}} \sum_{t=1}^{\mathcal{T}} M^t(x, y) \quad (2)$$

Pleiss et al. (2020) show that examples with low AUMs are ambiguous or tend to be mislabeled, and removing these examples can help the generalization performance. The vanilla AUM procedure can be summarized as follows: **1)** Train a classifier and monitor the AUM of each training example; **2)** Examples from the training set which have an AUM smaller than a threshold are considered mislabeled, hence are completely eliminated from the training set; and **3)** Train a new classifier on the filtered training set.

3.2 Proposed Approach

AUM-ST is a novel SSL approach that leverages the training dynamics of *unlabeled* examples to improve a model's performance. Algorithm 1 gives an overview of our AUM-ST. We first train a teacher model on weakly augmented labeled examples (Step 1) and use the trained teacher to make predictions and generate hard pseudo-labels for weakly augmented unlabeled examples (Step 2). Next, we monitor the training dynamics of these unlabeled data and their pseudo-labels (Step 3). Specifically, we characterize the unlabeled examples according to their contribution to model learning and generalization using AUM. Next, we filter out data with low AUM, since these examples are likely to hurt the generalization performance (Step 4). Then, we train a student model to be consistent with the teacher's predictions on unlabeled examples. Concretely, we train our student to minimize the combined cross-entropy on weakly augmented labeled examples and strongly-augmented, high-AUM unlabeled examples (Step 5). Finally, we

Algorithm 1 Proposed AUM-ST

Require: Labeled data $L = \{(x_1, y_1), (x_2, y_2), ...(x_n, y_n)\}$, unlabeled data $U = \{\hat{x}_1, \hat{x}_2, ...\hat{x}_m\}$ and γ AUM threshold.

1: Learn teacher model θ^t on weakly noised labeled data minimizing the following cross entropy loss
$L_{\theta^t} = \frac{1}{n}\sum_{i=1}^{n} H(y_i, p(y|\pi(x_i); \theta^t))$

2: Use the weakly noised teacher model to generate hard pseudo labels for weakly augmented unlabeled examples
$\hat{y}_i = \text{argmax}(p(y|\pi(\hat{x}_i); \theta^t)), \forall i = 1, \cdots, m$

3: Train model θ^{AUM} on weakly augmented training and unlabeled examples, and monitor the training dynamics of unlabeled examples over $\mathcal{T}$ epochs
$AUM(\hat{x}_i, \hat{y}_i) = \frac{1}{\mathcal{T}}\sum_{1}^{\mathcal{T}}[z_{\hat{y}_i} - max_{\hat{y}_i!=j}(z_j)]$,
where $z_{\hat{y}_i}$ and z_j are the logits corresponding to the pseudolabel $\hat{y}_i$ and the largest other logit produced by θ^{AUM}

4: Rank and select high-AUM unlabeled examples
$U^{AUM} = \{(\hat{x}_i, \hat{y}_i) \in U \quad | \quad AUM(\hat{x}_i, \hat{y}_i) > \gamma \}$,

5: Train a student model θ^s which minimizes the cross-entropy loss on weakly augmented labeled examples and strongly augmented high-AUM unlabeled examples.
$L_{\theta^s} = \frac{1}{n}\sum_{i=1}^{n} H(y_i, p(y|\pi(x_i); \theta^s)) + \frac{1}{k}\sum_{i=1}^{k} H(\hat{y}_i, p(y|\Pi(\hat{x}_i); \theta^s))$,
where k is the size of U^{AUM}

6: Use the student as a teacher and go back to Step 2

use the student as the new teacher and reiterate the process from Step 2. We also explored reusing the teacher (i.e., using a student initialized using the teacher network weights) to estimate the AUMs of unlabeled examples in Step 3 of our algorithm. However, we noticed a slight decrease in accuracy of 0.4%.

The main improvement of AUM-ST lies in the use of training dynamics to assess the quality of pseudo-labels of unlabeled examples. Based on this quality measure, AUM-ST successfully filters harmful pseudo-label noise and improves model performance. Using a strongly noised student (Step 5) is another important factor in our framework. We train our student to match the predictions on strongly augmented examples $\Pi(\hat{x}_i)$ to the teacher's predictions on weakly augmented examples $\pi(\hat{x}_i)$. The intuition of this design choice comes from recent work on semi-supervised learning in vision (Sohn et al., 2020; Zhang et al., 2021), which has successfully showed that this combination of weak and strong augmentations work extremely well in practice. Specifically, using weak augmentations to generate the pseudo-labels and computing the loss against strong augmentations enforces a type of consistency regularization that in our setup exposes the student to a more difficult environment, which leads to the student outperforming the teacher.

We use various approaches to obtain strongly augmented data. In our setup, our weak augmentations are created using synonym replacement (Kolomiyets et al., 2011) or SwitchOut (Wang et al., 2018), and strong augmentations are obtained by randomly performing Backtranslations using long chain lengths (> 5), SwitchOut and synonym replacements. We further discuss the impact of different types of augmentations in §5.4.

Other Factors AUM-ST works better in practice using various additional factors: **1)** Consistent with other state-of-the-art SSL frameworks (Sohn et al., 2020; Xie et al., 2020a,b), in Step 2 of the algorithm, we select unlabeled examples only if the teacher confidence is higher than a threshold value (i.e., 0.7 in AUM-ST). **2)** In Step 3, we train θ^{AUM} only on a subset of the unlabeled examples that pass the filtering from the previous step. While considering all the unlabeled examples might work when the labeled data is abundant, training a model with very few labeled examples and **a lot** of potentially noisy pseudo-labeled examples produces poor AUM estimations. **3)** We always balance the class distribution of the unlabeled examples. **4)** When training our model on both labeled and unlabeled examples (Step 5), our batches contain both labeled and unlabeled examples. The ratio of labeled to unlabeled examples is constant across all batches and set to $1:7$ (i.e., each batch contains seven times as many unlabeled examples as labeled examples).

4 Experiments and Results

In this section, we first introduce the eight benchmark text classification datasets used to evaluate AUM-ST (§4.1). Second, we introduce weak and

strong baselines (§4.2) which we compare against our AUM-ST. Next, we detail our experimental setup (§4.3) and conclude by presenting the results obtained on all datasets in low data regimes (§4.4).

4.1 Datasets

We consider various text classification datasets to benchmark our AUM-ST self-training approach. We first experiment with the Stanford Sentiment Treebank (SST) (Socher et al., 2013). SST contains $11,855$ sentences from movie reviews, annotated with five sentiment labels: *negative*, *somewhat negative*, *neutral*, *somewhat positive*, and *positive*. First, we consider the binarized version of the SST dataset, called **SST-2**, where the examples with the *negative* and *somewhat negative* labels are merged into a *negative* class, and the examples with the *somewhat positive* and *positive* labels are merged into a *positive* class (with neutral class being removed). Second, we consider the fine-grained version **SST-5**, which uses all five labels. Next, we consider the **IMDB** (Maas et al., 2011) movie reviews dataset. While the SST dataset is annotated at sentence level, an important particularity of the IMDB dataset is that it is annotated at review level, containing significantly longer text sequences.

GoEmotions (Demszky et al., 2020) is a sentence-level multi-label dataset created using Reddit comments. Containing more than $58,000$ sentences annotated with 27 emotions and the neutral class, GoEmotions provides a great opportunity to study the expression of fine-grained emotions and to develop emotion classification models. We experiment both with the highly granular version of the dataset (27 emotions and the neutral class, denoted by **GoEmotions-28** in our experiments) and the version of the dataset where the labels are clustered into the Ekman basic set of six emotions, namely anger, disgust, fear, joy, sadness, and surprise, denoted by **GoEmotions-Ek**. **CancerEmo** (Sosea and Caragea, 2020a) is a dataset annotated at sentence level with the eight basic Plutchik-8 (Plutchik, 1980) emotions. The data is collected from a cancer forum from an Online Health Community and contains $8,500$ total examples annotated with fine-grained emotions and $16,500$ sentences that express no emotions (the neutral class).

We also consider the task of question classification and experiment with **TREC-6** , a dataset of 5452 examples where fact-based questions are divided into six broad semantic categories. Finally, we test on the **Corpus of Linuistic Acceptability (CoLA)**, a dataset composed of 10657 sentences from 23 linguistics publications, manually annotated by expert linguists for acceptability (i.e., grammaticality).

4.2 Baselines

Weak Baselines In this section, we present two weak self-training baselines, where we experiment with various approaches of selecting what unlabeled data to use during self-training (i.e., Step 4 in our AUM-ST). Our first approach, entitled RAND, chooses the unlabeled set of examples to use during training at random. The second method considered is CONF, which selects unlabeled examples only if the model confidence passes a pre-defined threshold.

Strong Baselines First, we experiment with Uncertainty-aware Self-training (UST) (Mukherjee and Awadallah, 2020a) as a strong baseline. UST incorporates uncertainty estimates into the standard self-training framework by adding a few highly effective changes. UST computes uncertainty estimates for all unlabeled examples by stochastically passing the examples from this set through the model multiple times, with dropout enabled before each layer. The approach subsequently uses these uncertainty estimates to select what unlabeled data to use. Concretely, the model not only favors unlabeled data where the teacher model is confident, but also enforces low entropy of the teacher predictions. Second, we experiment with UDA (Xie et al., 2020a). UDA leverages Backtranslation (Edunov et al., 2018) and uses a consistency loss to enforce the model predictions on unlabeled data to be invariant to input noise.

4.3 Experimental Setup

We evaluate the performance of our AUM-ST by varying the number of training examples on the eight text classification benchmark datasets presented above. On each dataset, we experiment with 20, 50, 100, and 200 examples per class, which we sample without replacement. The remaining examples are used as unlabeled data. We follow the exact evaluation metrics used in the works introducing the datasets: accuracy for SST-2, SST-5, IMDB, and TREC-6, macro F1 for GoEmotions and CancerEmo, and Matthews correlation for CoLA. In each setup, we also run our models five times, with different parameter intializations, and report the average results, as

Dataset (Metric)	SST-2 (Accuracy)				SST-5 (Accuracy)			
Num Labels	20 lb/cl	50 lb/cl	100 lb/cl	200 lb/cl	20 lb/cl	50 lb/cl	100 lb/cl	200 lb/cl
BASE	$78.0_{\pm 5.56}$	$80.1_{\pm 4.16}$	$83.5_{\pm 3.48}$	$84.6_{\pm 3.29}$	$34.3_{\pm 4.52}$	$41.4_{\pm 3.51}$	$41.8_{\pm 2.45}$	$42.0_{\pm 1.52}$
RAND	$79.1_{\pm 5.15}$	$81.4_{\pm 4.01}$	$83.7_{\pm 3.27}$	$85.9_{\pm 3.06}$	$34.5_{\pm 3.98}$	$42.1_{\pm 3.21}$	$43.4_{\pm 2.14}$	$43.1_{\pm 1.50}$
CONF	$83.5_{\pm 4.51}$	$82.4_{\pm 3.61}$	$87.2_{\pm 3.55}$	$88.0_{\pm 3.18}$	$35.6_{\pm 5.11}$	$42.2_{\pm 4.61}$	$42.6_{\pm 1.52}$	$43.5_{\pm 1.37}$
UST	$84.3_{\pm 3.15}$	$86.3_{\pm 3.24}$	$89.1_{\pm 2.91}$	$91.2_{\pm 1.83}$	$36.4_{\pm 3.61}$	$44.2_{\pm 2.55}$	$45.6_{\pm 1.49}$	$46.1_{\pm 1.23}$
UDA	$\mathbf{84.8_{\pm 2.83}}$	$91.3_{\pm 2.58}$	$91.3_{\pm 2.29}$	$92.4_{\pm 1.63}$	$\mathbf{37.5_{\pm 2.57}}$	$45.1_{\pm 2.17}$	$45.9_{\pm 1.42}$	$47.2_{\pm 1.11}$
AUM-ST	$82.4_{\pm 3.12}$	$\mathbf{92.9_{\pm 2.41}}$	$\mathbf{93.1_{\pm 2.17}}$	$\mathbf{93.2_{\pm 1.52}}$	$37.4_{\pm 2.17}$	$\mathbf{46.2_{\pm 2.06}}$	$\mathbf{47.1_{\pm 1.37}}$	$\mathbf{47.9_{\pm 1.05}}$
Dataset (Metric)	IMDB (Accuracy)				GoEmotions-28 (F1)			
Num Labels	20 lb/cl	50 lb/cl	100 lb/cl	200 lb/cl	20 lb/cl	50 lb/cl	100 lb/cl	200 lb/cl
BASE	$69.1_{\pm 3.11}$	$78.4_{\pm 2.87}$	$75.3_{\pm 1.93}$	$80.3_{\pm 1.76}$	$09.2_{\pm 4.18}$	$20.4_{\pm 3.57}$	$26.3_{\pm 2.17}$	$21.4_{\pm 2.52}$
RAND	$67.1_{\pm 3.53}$	$79.4_{\pm 2.99}$	$76.4_{\pm 2.34}$	$81.4_{\pm 1.87}$	$12.5_{\pm 4.37}$	$25.6_{\pm 3.81}$	$26.9_{\pm 2.61}$	$20.5_{\pm 2.75}$
CONF	$71.1_{\pm 3.35}$	$80.5_{\pm 3.04}$	$76.5_{\pm 2.15}$	$81.5_{\pm 1.75}$	$15.4_{\pm 3.81}$	$25.3_{\pm 3.56}$	$24.1_{\pm 2.45}$	$21.9_{\pm 2.39}$
UST	$72.5_{\pm 2.81}$	$85.4_{\pm 2.91}$	$74.9_{\pm 2.05}$	$82.7_{\pm 1.53}$	$17.4_{\pm 3.31}$	$23.1_{\pm 3.41}$	$24.6_{\pm 2.15}$	$31.5_{\pm 2.17}$
UDA	$77.4_{\pm 2.55}$	$86.5_{\pm 2.76}$	$79.7_{\pm 1.78}$	$83.7_{\pm 1.37}$	$24.5_{\pm 2.61}$	$25.9_{\pm 2.73}$	$26.4_{\pm 1.75}$	$30.9_{\pm 1.63}$
AUM-ST	$\mathbf{81.7_{\pm 2.17}}$	$\mathbf{88.3_{\pm 2.55}}$	$\mathbf{82.3_{\pm 3.05}}$	$\mathbf{85.3_{\pm 1.62}}$	$\mathbf{25.1_{\pm 2.51}}$	$\mathbf{26.1_{\pm 2.67}}$	$\mathbf{28.1_{\pm 1.53}}$	$\mathbf{31.9_{\pm 1.57}}$
Dataset (Metric)	GoEmotions-Ek (F1)				CancerEmo (F1)			
Num Labels	20 lb/cl	50 lb/cl	100 lb/cl	200 lb/cl	20 lb/cl	50 lb/cl	100 lb/cl	200 lb/cl
BASE	$20.3_{\pm 4.56}$	$25.7_{\pm 4.25}$	$34.2_{\pm 3.59}$	$56.2_{\pm 2.38}$	$23.1_{\pm 4.51}$	$34.2_{\pm 3.53}$	$41.5_{\pm 3.05}$	$51.3_{\pm 2.46}$
RAND	$20.6_{\pm 4.37}$	$25.4_{\pm 4.02}$	$33.9_{\pm 3.29}$	$56.9_{\pm 2.31}$	$24.1_{\pm 4.61}$	$35.2_{\pm 3.71}$	$43.6_{\pm 2.96}$	$50.4_{\pm 2.37}$
CONF	$20.5_{\pm 4.06}$	$25.9_{\pm 3.57}$	$34.7_{\pm 3.17}$	$58.8_{\pm 2.19}$	$27.3_{\pm 4.37}$	$38.1_{\pm 3.64}$	$45.7_{\pm 2.74}$	$53.4_{\pm 2.18}$
UST	$21.5_{\pm 3.57}$	$26.4_{\pm 3.26}$	$39.5_{\pm 3.19}$	$59.1_{\pm 2.45}$	$29.1_{\pm 3.94}$	$41.2_{\pm 3.21}$	$49.6_{\pm 1.82}$	$56.4_{\pm 1.75}$
UDA	$23.1_{\pm 3.01}$	$27.8_{\pm 2.81}$	$40.1_{\pm 2.54}$	$59.9_{\pm 2.17}$	$31.2_{\pm 3.59}$	$44.3_{\pm 3.07}$	$50.3_{\pm 1.67}$	$56.1_{\pm 1.43}$
AUM-ST	$\mathbf{24.5_{\pm 2.87}}$	$\mathbf{28.9_{\pm 2.85}}$	$\mathbf{40.5_{\pm 2.73}}$	$\mathbf{62.2_{\pm 2.19}}$	$\mathbf{31.4_{\pm 3.48}}$	$\mathbf{46.2_{\pm 3.12}}$	$\mathbf{51.3_{\pm 1.73}}$	$\mathbf{56.9_{\pm 1.38}}$
Dataset (Metric)	TREC-6 (Accuracy)				COLA (Matthew Correlation)			
Num Labels/Class	20 lb/cl	50 lb/cl	100 lb/cl	200 lb/cl	20 lb/cl	50 lb/cl	100 lb/cl	200 lb/cl
BASE	$80.4_{\pm 2.57}$	$85.1_{\pm 2.31}$	$83.3_{\pm 1.87}$	$85.9_{\pm 1.24}$	$25.3_{\pm 3.93}$	$29.7_{\pm 3.31}$	$32.1_{\pm 2.76}$	$43.9_{\pm 2.41}$
RAND	$80.1_{\pm 2.87}$	$84.8_{\pm 2.53}$	$84.9_{\pm 2.23}$	$83.2_{\pm 1.35}$	$24.5_{\pm 5.21}$	$27.4_{\pm 3.67}$	$33.3_{\pm 3.26}$	$44.1_{\pm 2.99}$
CONF	$81.3_{\pm 2.76}$	$86.2_{\pm 2.15}$	$89.2_{\pm 2.09}$	$84.5_{\pm 1.08}$	$27.7_{\pm 3.65}$	$29.6_{\pm 3.25}$	$34.1_{\pm 2.65}$	$46.7_{\pm 2.83}$
UST	$82.2_{\pm 2.56}$	$87.5_{\pm 2.08}$	$89.3_{\pm 2.31}$	$85.9_{\pm 1.15}$	$28.6_{\pm 3.85}$	$31.8_{\pm 3.46}$	$37.5_{\pm 2.55}$	$48.6_{\pm 2.67}$
UDA	$84.2_{\pm 2.35}$	$88.1_{\pm 2.02}$	$89.7_{\pm 1.41}$	$\mathbf{91.6_{\pm 0.95}}$	$30.7_{\pm 3.28}$	$34.6_{\pm 2.45}$	$49.2_{\pm 2.41}$	$54.2_{\pm 2.17}$
AUM-ST	$\mathbf{84.9_{\pm 2.39}}$	$\mathbf{88.5_{\pm 1.89}}$	$\mathbf{90.3_{\pm 1.45}}$	$91.4_{\pm 0.99}$	$\mathbf{32.1_{\pm 3.17}}$	$\mathbf{35.4_{\pm 2.48}}$	$\mathbf{50.6_{\pm 2.35}}$	$\mathbf{55.4_{\pm 2.12}}$

Table 1: Results on eight text classification benchmarks and various low data regime setups.

well as their standard deviations. Model-wise, all our experiments use the BERT (Devlin et al., 2019) base uncased as the backbone model and the HuggingFace Transformers (Wolf et al., 2020) library for the implementation. We use the AUM package provided by Pleiss et al. (2020) for the AUM estimation of unlabeled examples. We use the translation models provided by Tiedemann and Thottingal (2020) for backtranslation. We present the hyperparametrs of our model in Appendix A.

4.4 Results

We show the results obtained across the eight datasets in Table 1. Overall, we note that our approach is extremely effective, significantly outperforming strong baselines on all datasets.

For example, AUM-ST pushes the accuracy over the strongest UDA baseline by 1.6% on SST-2 and 1.1% on SST-5 using 50 labels per class. Critically, using 100 examples per class on SST-2, AUM-ST obtains 93.1% accuracy, a considerable improvement of 9.6% over the baseline BERT model. Remarkably, on IMDB, we improve the accuracy over UDA by 4.3% with 20 examples per class and over the fully supervised BERT by 12.6%. We see consistent improvements on both GoEmotions-28 and GoEmotions-Ek datasets as well, where our method is particularly effective using 100 and 200 examples per class. Notably, our AUM-ST improves upon the baseline BERT model by 10% in F1 score using 200 examples per class on GoEmotions-28, and pushes the F1 score over UDA by 2.3% using the same amount of examples on the GoEmotions-Ek dataset.

We notice that on the TREC-6 question classification dataset, the UDA model slightly outperforms our AUM-ST using 200 examples per class, but lags behind in the other setups. For example, AUM-ST pushes the accuracy by 0.7% using 20 examples per class. Results on COLA also showcase

Num Labels	20 lb/cl	50 lb/cl	100 lb/cl	200 lb/cl	20 lb/cl	50 lb/cl	100 lb/cl	200 lb/cl
Dataset (Metric)	IMDB (Accuracy)				GoEmotions-28 (F1)			
UDA-AUG	$78.1_{\pm 2.55}$	$86.0_{\pm 2.76}$	$81.2_{\pm 1.78}$	$83.4_{\pm 1.37}$	$24.3_{\pm 2.61}$	$25.5_{\pm 2.73}$	$26.2_{\pm 1.75}$	$30.7_{\pm 1.63}$
AUM-ST	$\mathbf{81.7_{\pm 2.17}}$	$\mathbf{88.3_{\pm 2.55}}$	$\mathbf{82.3_{\pm 3.05}}$	$\mathbf{85.3_{\pm 1.62}}$	$\mathbf{25.1_{\pm 2.51}}$	$\mathbf{26.1_{\pm 2.67}}$	$\mathbf{28.1_{\pm 1.53}}$	$\mathbf{31.9_{\pm 1.57}}$
Dataset (Metric)	TREC-6 (Accuracy)				COLA (Matthew Correlation)			
UDA-AUG	$84.0_{\pm 2.35}$	$87.5_{\pm 2.02}$	$89.8_{\pm 1.41}$	$\mathbf{91.8_{\pm 0.95}}$	$31.2_{\pm 3.28}$	$34.4_{\pm 2.45}$	$49.9_{\pm 2.41}$	$54.1_{\pm 2.17}$
AUM-ST	$\mathbf{84.9_{\pm 2.39}}$	$\mathbf{88.5_{\pm 1.89}}$	$\mathbf{90.3_{\pm 1.45}}$	$91.4_{\pm 0.99}$	$\mathbf{32.1_{\pm 3.17}}$	$\mathbf{35.4_{\pm 2.48}}$	$\mathbf{50.6_{\pm 2.35}}$	$\mathbf{55.4_{\pm 2.12}}$

Table 2: Ablation study of our AUM-ST.

the effectiveness of our methods, where we see an improvement of 1.4% in Matthews correlation over the strong UDA baseline and 6.8% over the fully supervised approach using 20 examples per class. These results indicate that AUM-ST performs extremely well in low data regimes, and can be used effectively when training data is scarce. To this end, we emphasize that AUM-ST can considerably mitigate the annotation efforts needed to obtain good performance on the task at hand.

5 Analysis

5.1 Ablation Study

While our augmentation techniques play an important part in our AUM-ST, we argue that the AUM filtering is a vital component of our framework. To this end, we perform an ablation study to verify that the improvements in performance do not come solely from the consistency loss of weak and strong augmentations. To this end, we retrain our strongest baseline UDA (Xie et al., 2020a) in all data regimes (20/50/100/200 labels per class) using the same augmentations as our AUM-ST on four datasets: IMDB, GoEmotions, TREC-6 and COLA. Concretely, this variation of UDA (denoted by UDA-AUG) minimizes the KL divergence between the predictions on weakly augmented unlabeled examples and the predictions on strongly augmented unlabeled examples. We show the results obtained in Table 2 where we observe that AUM-ST obtains steady improvements in performance over UDA of 1.2% on average. Critically, these results show that the improvements in performance come from our AUM-based filtering method, emphasizing its effectiveness.

5.2 AUM-ST when large labeled data is available

We evaluate AUM-ST in high-resource settings to verify if it performs well on large datasets. To this end, we first seek to collect additional unlabeled data to use alongside the provided training sets.

Unlabeled Data We collect in-domain unlabeled data (if not provided) for six out of the eight datasets presented previously: **SST-2** and **SST-5**: We use the Kaggle Rotten Tomatoes corpus which contains more than one million reviews. We employ the same preprocessing techniques as in the original paper (Socher et al., 2013) (e.g., splitting at sentence level). **IMDB**: The IMDB dataset (Maas et al., 2011) already contains an unlabeled set of examples provided by the authors, hence we use it in our experiments. **CancerEmo**: We use the same discussion boards from the Cancer Survivors Network used in the work introducing the dataset (Sosea and Caragea, 2020b). **GoEmotions-28** and **GoEmotions-Ek**: Since the authors do not disclose the subreddits used for sampling their data, we resort to using a general Reddit dump (Henderson et al., 2019). We omit TREC-6 and CoLA in this experiment since additional unlabeled data for gramaticality or question classification is hard to find.

To enable reproducibility and spur further research into SSL techniques for text classification, we will make the collected unlabeled data available to the research community.

Results We show the results obtained in high-resource settings in Table 3. First, we observe that on SST-2, SST-5, and IMDB our weak baselines with unlabeled data do not bring any improvements over the fully supervised approach. Second, interestingly, while UST outperforms the base model on the SST-2 and IMDB datasets, the performance on the fine-grained SST-5 is extremely low. Finally, our AUM-ST is successful on all the datasets, improving upon the supervised model by 1.4% on average and outperforming all the strong baselines.

On GoEmotions-28 and GoEmotions-Ek (Demszky et al., 2020), our weak baselines, RAND and CONF marginally outperform the baseline BERT, improving the average F1 by 0.1% and 0.3%, respectively. Interestingly, UST performs poorly on this dataset, being outperformed by the trivial CONF

	BASE	RAND	CONF	UST	UDA	AUM
SST-2	$92.2_{\pm 1.56}$	$91.1_{\pm 1.65}$	$92.1_{\pm 1.42}$	$93.0_{\pm 1.37}$	$92.8_{\pm 1.29}$	$\mathbf{93.5}_{\pm \mathbf{1.35}}$
SST-5	$53.2_{\pm 2.49}$	$50.3_{\pm 2.42}$	$51.7_{\pm 2.39}$	$52.7_{\pm 2.19}$	$54.1_{\pm 2.07}$	$\mathbf{54.8}_{\pm \mathbf{2.09}}$
IMDB	$93.7_{\pm 0.93}$	$93.5_{\pm 0.96}$	$93.6_{\pm 0.97}$	$94.2_{\pm 0.87}$	$93.9_{\pm 0.88}$	$\mathbf{95.1}_{\pm \mathbf{0.86}}$
GoEmotions-28	$46.2_{\pm 1.25}$	$46.3_{\pm 1.18}$	$46.5_{\pm 1.15}$	$46.3_{\pm 1.19}$	$46.9_{\pm 1.25}$	$\mathbf{47.7}_{\pm \mathbf{1.21}}$
GoEmotions-Ek	$67.7_{\pm 1.02}$	$67.9_{\pm 1.35}$	$68.2_{\pm 1.29}$	$68.1_{\pm 0.87}$	$68.9_{\pm 0.84}$	$\mathbf{69.5}_{\pm \mathbf{0.85}}$
CancerEmo	$70.1_{\pm 1.34}$	$70.3_{\pm 1.37}$	$70.5_{\pm 1.31}$	$71.3_{\pm 1.27}$	$72.5_{\pm 0.98}$	$\mathbf{73.2}_{\pm \mathbf{1.05}}$

Table 3: Comparison of different self-training methods using the entire training set and additional unlabeled data. We report the results in terms of accuracy on SST-2, SST-5, IMDB and macro F1 on GoEmotions-28, GoEmotions-Ek, and CancerEmo.

approach. We note that UDA performs the best among the baselines. However, our AUM-ST consistently outperforms other methods, and yields a 1.5% improvement in F1 score over the supervised model. On CancerEmo (Sosea and Caragea, 2020a), we note that UST performs much better with an improvement in F1 of 1.2% over the supervised classifier. Our AUM-ST approach is still the most successful, with a considerable 3.1% improvement over the baseline BERT, and 0.7% improvement over the strong UDA model.

While AUM-ST is particularly effective in low resource settings (as shown in §4.4), the results here also showcase the feasibility of our approach, which consistently outperforms all the other methods both in low-resource settings and high-resource (large labeled data) settings.

5.3 Unlabeled Data Impurity

As mentioned previously, noisy pseudo-labels can be detrimental to learning effective SSL models. In this section, we analyze the unlabeled data impurity (i.e., the fraction of unlabeled data which is incorrectly classified by our model) to compare the pseudo-label quality of various SSL methods against our AUM-ST. We emphasize that an effective SSL approach should aim to minimize impurity; low impurity indicates that the pseudo-labels of the teacher network are of high quality. We perform this analysis on the GoEmotions-28 dataset in a low data regime, using 200 examples per class. In this setup, since the unlabeled data is created from the original (labeled) training set, we can easily compute the unlabeled error rate. We show the impurity at the end of each self training iteration in Figure 1. Notably, at the end of the training process, AUM-ST improves the impurity by 1.4% over the UDA model, and by 3% over the UST method. Interestingly, we observe that the methods perform on-par with each other until Iteration 10, when the impurity of AUM-ST becomes lower than the other methods.

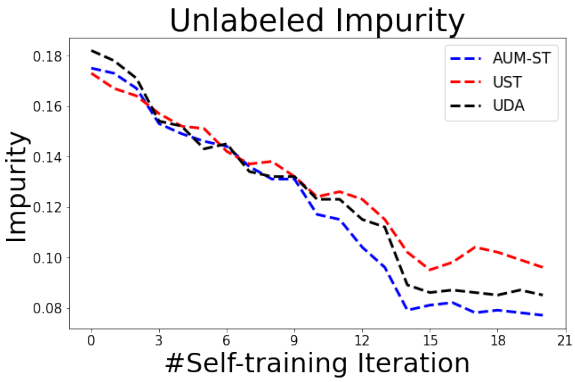

Figure 1: Comparison of impurity between the UST, UDA, and our AUM-ST model on the GoEmotions-28 dataset with 200 examples per class.

5.4 The Impact of Weak and Strong Augmentations

In this section, we analyze how our model performs when trained under various weak augmentations (π) and strong augmentations (Π) in our self-training framework. We show in Table 4 the performance in terms of macro F1 of AUM-ST using various combinations of π and Π on the GoEmotions-28 dataset with 200 examples per class. Note that we experiment with every combination of π and Π (even combinations when π is a stronger augmentation than Π) in order to also analyze the behavior of our approach when using stronger augmentations to generate the teacher pseudo-labels. We consider in our analysis the following augmentation strategies: no augmentation (NoAug), synonym replacement (SynRepl) (Kolomiyets et al., 2011), SwitchOut (Wang et al., 2018), and BT-n, which denotes Backtranslation (Edunov et al., 2018) with a chain of length n and languages such as German, French and Italian. We also consider a combination of these augmentation strategies (e.g., backtranslation, synonym replacement, and SwitchOut).

Interestingly, we can see from the table that using SwitchOut as π and a combination of Backtranslation with large n, Synonym Replacement and SwitchOut as Π yields the best results, improv-

		Π							
		NoAug	SynRepl	SwitchOut	BT-1	BT-5	BT-10	BT-10 + SynRepl	BT-10 + SynRepl + SwitchOut
π	NoAug	-	0.232	0.239	0.245	0.249	0.278	0.291	0.305
	SynRepl	0.216	0.245	0.251	0.264	0.278	0.288	0.314	0.316
	SwitchOut	0.223	0.221	0.231	0.248	0.255	0.289	0.311	**0.319**
	BT-1	0.231	0.235	0.251	0.267	0.284	0.289	0.293	0.301
	BT-5	0.201	0.267	0.278	0.289	0.291	0.292	0.295	0.294
	BT-10	0.231	0.278	0.282	0.285	0.283	0.274	0.289	0.283
	BT-10 + SynRepl	0.245	0.254	0.253	0.279	0.283	0.285	0.293	0.291
	BT-10 + SynRepl + SwitchOut	0.222	0.241	0.247	0.265	0.273	0.285	0.281	0.286

Table 4: Performance using various weak augmentations π and strong augmentations Π on the GoEmotions-28 dataset using 200 examples per class.

ing upon the fully supervised model by as much as 10% in F1. We can also see from the table that, as π goes from weak augmentations to strong augmentations, the performance degrades compared to using low-noise weak augmentations. These results emphasize the importance of both weak and strong data augmentations in our AUM-ST, indicating that it is a vital component of our framework.

5.5 Computational Costs

In this section, we discuss the computational cost of our AUM-ST and how it compares with other methods. First, we note that AUM-ST trains an additional model compared to other teacher-student approaches such as CONF or UST to perform the AUM estimation (Step 3 of our algorithm). However, the computational costs incurred by this additional training step are not a serious issue in low resource settings. Even in setups with large amounts of both labeled and unlabeled data, our computational cost is not significantly higher than the other methods because our AUM-ST method converges in a lower number of steps despite that it encompasses an additional training stage. Concretely, AUM-ST is 15% more computationally expensive than the traditional pseudo-labeling (i.e., the CONF method). Moreover AUM-ST converges three times faster compared to UST (Mukherjee and Awadallah, 2020b) and twice as fast as UDA (Xie et al., 2020a).

6 Conclusion

We improve the traditional self-training framework through a novel Area Under the Margin unlabeled example selection technique, and show that our approach is effective in a wide range of text classification tasks. We studied our approach in various domains (social networks, forums, online platforms) and contexts (movie reviews, medical forum discussions, fact-based questions), and observed that our AUM-ST outperforms other strong self-training approaches. In the future, we plan to incorporate other approaches of learning under label noise into SSL frameworks such as self-training.

Limitations

This work shows that achieving good performance in text classification with limited labeled data is possible. Unfortunately, this is possible exclusively if there is easy access to unlabeled data. Moreover, while unlabeled data for some tasks is hard to obtain (as we found for TREC and CoLA datasets), we also emphasize that even in the presence of unlabeled data, its distribution can be mismatched with the labeled data distribution, which was shown to be particulary challenging to deal with in SSL (Coates et al., 2011). Our work does not study this scenario, however, we aim to further explore our method in this setting.

Acknowledgements

This research is supported in part by NSF CAREER award #1802358, NSF CRI award #1823292, NSF IIS award #2107518, and UIC Discovery Partners Institute (DPI) award. Any opinions, findings, and conclusions expressed here are those of the authors and do not necessarily reflect the views of NSF or DPI. We thank AWS for computational resources. We also thank our anonymous reviewers for their constructive feedback.

References

Peter L. Bartlett, Dylan J. Foster, and Matus Telgarsky. 2017. Spectrally-normalized margin bounds for neural networks. *CoRR*, abs/1706.08498.

David Berthelot, Nicholas Carlini, Ekin D. Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. 2019a. Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring. *arXiv preprint arXiv:1911.09785*.

David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin Raffel. 2019b. Mixmatch: A holistic approach to semi-supervised learning. *arXiv preprint arXiv:1905.02249*.

Carla E Brodley and Mark A Friedl. 1999. Identifying mislabeled training data. *Journal of artificial intelligence research*, 11:131–167.

Jiaao Chen, Zichao Yang, and Diyi Yang. 2020a. MixText: Linguistically-informed interpolation of hidden space for semi-supervised text classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2147–2157, Online. Association for Computational Linguistics.

Jiaao Chen, Zichao Yang, and Diyi Yang. 2020b. MixText: Linguistically-informed interpolation of hidden space for semi-supervised text classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2147–2157, Online. Association for Computational Linguistics.

Kevin Clark, Minh-Thang Luong, Christopher D. Manning, and Quoc Le. 2018. Semi-supervised sequence modeling with cross-view training. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1914–1925, Brussels, Belgium. Association for Computational Linguistics.

Adam Coates, Andrew Ng, and Honglak Lee. 2011. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223. JMLR Workshop and Conference Proceedings.

Dorottya Demszky, Dana Movshovitz-Attias, Jeongwoo Ko, Alan Cowen, Gaurav Nemade, and Sujith Ravi. 2020. Goemotions: A dataset of fine-grained emotions. *arXiv preprint arXiv:2005.00547*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. 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 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *Conference of the Association for Computational Linguistics (ACL)*.

Gamaleldin Fathy Elsayed, Dilip Krishnan, Hossein Mobahi, Kevin Regan, and Samy Bengio. 2018. Large margin deep networks for classification.

Jacob Goldberger and Ehud Ben-Reuven. 2016. Training deep neural-networks using a noise adaptation layer.

Chen Gong, Dacheng Tao, Stephen J. Maybank, Wei Liu, Guoliang Kang, and Jie Yang. 2016. Multi-modal curriculum learning for semi-supervised image classification. *IEEE Transactions on Image Processing*, 25(7):3249–3260.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 1321–1330. JMLR.org.

Suchin Gururangan, Tam Dang, Dallas Card, and Noah A Smith. 2019a. Variational pretraining for semi-supervised text classification. *arXiv preprint arXiv:1906.02242*.

Suchin Gururangan, Tam Dang, Dallas Card, and Noah A. Smith. 2019b. Variational pretraining for semi-supervised text classification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5880–5894, Florence, Italy. Association for Computational Linguistics.

Matthew Henderson, Paweł Budzianowski, Iñigo Casanueva, Sam Coope, Daniela Gerz, Girish Kumar, Nikola Mrkšić, Georgios Spithourakis, Pei-Hao Su, Ivan Vulić, and Tsung-Hsien Wen. 2019. A repository of conversational datasets. In *Proceedings of the First Workshop on NLP for Conversational AI*, pages 1–10, Florence, Italy. Association for Computational Linguistics.

Pavel Izmailov, Polina Kirichenko, Marc Finzi, and Andrew Gordon Wilson. 2020. Semi-supervised learning with normalizing flows. In *International Conference on Machine Learning*, pages 4615–4630. PMLR.

Yiding Jiang, Dilip Krishnan, Hossein Mobahi, and Samy Bengio. 2018. Predicting the generalization gap in deep networks with margin distributions.

Hoel Kervadec, Jose Dolz, Eric Granger, and Ismail Ben Ayed. 2019. Curriculum semi-supervised segmentation. *CoRR*, abs/1904.05236.

Oleksandr Kolomiyets, Steven Bethard, and Marie-Francine Moens. 2011. Model-portability experiments for textual temporal analysis. In *Proceedings of the 49th Annual Meeting of the Association for*

Computational Linguistics: Human Language Technologies: Short Papers - Volume 2, HLT '11, page 271–276, USA. Association for Computational Linguistics.

Samuli Laine and Timo Aila. 2017. Temporal ensembling for semi-supervised learning. In *ICLR (Poster)*. OpenReview.net.

Dong-Hyun Lee et al. 2013. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, page 896.

Yang Liu and Hongyi Guo. 2020. Peer loss functions: Learning from noisy labels without knowing noise rates. In *International Conference on Machine Learning*, pages 6226–6236. PMLR.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Geoffrey J McLachlan. 1975. Iterative reclassification procedure for constructing an asymptotically optimal rule of allocation in discriminant analysis. *Journal of the American Statistical Association*, 70(350):365–369.

Takeru Miyato, Andrew M Dai, and Ian Goodfellow. 2016. Adversarial training methods for semi-supervised text classification. *arXiv preprint arXiv:1605.07725*.

Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. 2018. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1979–1993.

Subhabrata Mukherjee and Ahmed Awadallah. 2020a. Uncertainty-aware self-training for few-shot text classification. In *Advances in Neural Information Processing Systems*, volume 33, pages 21199–21212. Curran Associates, Inc.

Subhabrata Mukherjee and Ahmed Hassan Awadallah. 2020b. Uncertainty-aware self-training for text classification with few labels. *arXiv preprint arXiv:2006.15315*.

Subhabrata Mukherjee and Ahmed Hassan Awadallah. 2020. Uncertainty-aware self-training for few-shot text classification. In *Advances in Neural Information Processing Systems (NeurIPS 2020)*, Online.

Geoff Pleiss, Tianyi Zhang, Ethan Elenberg, and Kilian Q Weinberger. 2020. Identifying mislabeled data using the area under the margin ranking. In *Advances in Neural Information Processing Systems*, volume 33, pages 17044–17056. Curran Associates, Inc.

Robert Plutchik. 1980. A general psychoevolutionary theory of emotion. In *Theories of emotion*, pages 3–33. Elsevier.

Antti Rasmus, Mathias Berglund, Mikko Honkala, Harri Valpola, and Tapani Raiko. 2015. Semi-supervised learning with ladder networks. In *Advances in neural information processing systems*, pages 3546–3554.

Chuck Rosenberg, Martial Hebert, and Henry Schneiderman. 2005. Semi-supervised self-training of object detection models.

Devendra Singh Sachan, Manzil Zaheer, and Ruslan Salakhutdinov. 2019. Revisiting lstm networks for semi-supervised text classification via mixed objective function. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6940–6948.

Mehdi Sajjadi, Mehran Javanmardi, and Tolga Tasdizen. 2016a. Mutual exclusivity loss for semi-supervised deep learning. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 1908–1912. IEEE.

Mehdi Sajjadi, Mehran Javanmardi, and Tolga Tasdizen. 2016b. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. *Advances in neural information processing systems*, 29:1163–1171.

Shreyas Saxena, Oncel Tuzel, and Dennis DeCoste. 2019. Data parameters: A new family of parameters for learning a differentiable curriculum. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Henry Scudder. 1965. Probability of error of some adaptive pattern-recognition machines. *IEEE Transactions on Information Theory*, 11(3):363–371.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. 2020. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In *Advances in Neural Information Processing Systems*, volume 33, pages 596–608. Curran Associates, Inc.

Tiberiu Sosea and Cornelia Caragea. 2020a. Cancer-emo: A dataset for fine-grained emotion detection. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8892–8904.

Tiberiu Sosea and Cornelia Caragea. 2020b. CancerEmo: A dataset for fine-grained emotion detection. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8892–8904, Online. Association for Computational Linguistics.

Antti Tarvainen and Harri Valpola. 2017. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in neural information processing systems*, pages 1195–1204.

Jörg Tiedemann and Santhosh Thottingal. 2020. OPUS-MT — Building open translation services for the World. In *Proceedings of the 22nd Annual Conferenec of the European Association for Machine Translation (EAMT)*, Lisbon, Portugal.

Vikas Verma, Kenji Kawaguchi, Alex Lamb, Juho Kannala, Arno Solin, Yoshua Bengio, and David Lopez-Paz. 2021. Interpolation consistency training for semi-supervised learning. *Neural Networks*.

Xinyi Wang, Hieu Pham, Zihang Dai, and Graham Neubig. 2018. SwitchOut: an efficient data augmentation algorithm for neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 856–861, Brussels, Belgium. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. 2020a. Unsupervised data augmentation for consistency training. In *Advances in Neural Information Processing Systems*, volume 33, pages 6256–6268. Curran Associates, Inc.

Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. 2020b. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10687–10698.

Diyi Yang, Jiaao Chen, Zichao Yang, Dan Jurafsky, and Eduard Hovy. 2019. Let's make your request more persuasive: Modeling persuasive strategies via semi-supervised neural nets on crowdfunding platforms. 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 3620–3630, Minneapolis, Minnesota. Association for Computational Linguistics.

Diyi Yang, Miaomiao Wen, and Carolyn Rosé. 2015. Weakly supervised role identification in teamwork interactions. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1671–1680, Beijing, China. Association for Computational Linguistics.

Zichao Yang, Zhiting Hu, Ruslan Salakhutdinov, and Taylor Berg-Kirkpatrick. 2017. Improved variational autoencoders for text modeling using dilated convolutions. *CoRR*, abs/1702.08139.

Qing Yu, Daiki Ikami, Go Irie, and Kiyoharu Aizawa. 2020. Multi-task curriculum framework for open-set semi-supervised learning. In *ECCV*.

Bowen Zhang, Yidong Wang, Wenxin Hou, HAO WU, Jindong Wang, Manabu Okumura, and Takahiro Shinozaki. 2021. Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. In *Advances in Neural Information Processing Systems*, volume 34, pages 18408–18419. Curran Associates, Inc.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. 2016. Understanding deep learning requires rethinking generalization. *CoRR*, abs/1611.03530.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. 2017a. Understanding deep learning requires rethinking generalization.

Yizhe Zhang, Dinghan Shen, Guoyin Wang, Zhe Gan, Ricardo Henao, and Lawrence Carin. 2017b. Deconvolutional paragraph representation learning. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

A Hyperparameters

In this section, we detail the hyperparameter search space and the final hyperparameters used by our model. For our final AUM-ST model, we use a learning rate of $5e-5$ and a variable batch size depending on the size of the training set; ranging from 8 for experiments with 20 examples per class, and 32 when using 200 examples per class. In terms of search space, we tried batches in the range $4 \rightarrow 64$ and learning rates in the range $1e-5 \rightarrow 9e-5$ with a step of $1e-5$. Training our AUM-ST model on our A5000 GPU takes on average ~3

hours to complete for each dataset in low resource settings.