Semi-Supervised Text Classification with Balanced Deep Representation Distributions

Changchun Li, Ximing Li[*†] and **Jihong Ouyang**[†]
College of Computer Science and Technology, Jilin University, China
Key Laboratory of Symbolic Computation and Knowledge Engineering
of Ministry of Education, China
changchunli93@gmail.com, liximing86@gmail.com, ouyj@jlu.edu.cn

Abstract

Semi-Supervised Text Classification (SSTC) mainly works under the spirit of self-training. They initialize the deep classifier by training over labeled texts; and then alternatively predict unlabeled texts as their pseudo-labels and train the deep classifier over the mixture of labeled and pseudo-labeled texts. Naturally, their performance is largely affected by the accuracy of pseudo-labels for unlabeled texts. Unfortunately, they often suffer from low accuracy because of the margin bias problem caused by the large difference between representation distributions of labels in SSTC. To alleviate this problem, we apply the angular margin loss, and perform Gaussian linear transformation to achieve balanced label angle variances, *i.e.,* the variance of label angles of texts within the same label. More accuracy of predicted pseudo-labels can be achieved by constraining all label angle variances balanced, where they are estimated over both labeled and pseudo-labeled texts during self-training loops. With this insight, we propose a novel SSTC method, namely Semi-Supervised Text Classification with Balanced Deep representation Distributions (S^{2}TC-BDD). To evaluate S^{2}TC-BDD, we compare it against the state-of-the-art SSTC methods. Empirical results demonstrate the effectiveness of S^{2}TC-BDD, especially when the labeled texts are scarce.

1 Introduction

Semi-**S**upervised **L**earning (**SSL**) refers to the paradigm of learning with labeled as well as unlabeled data to perform certain applications (van Engelen and Hoos, 2020). Especially, developing effective SSL models for classifying text data has long been a goal for the studies of natural language processing, because labeled texts are difficult to collect in many real-world scenarios. Formally, this research topic is termed as **S**emi-**S**upervised **T**ext **C**lassification (**SSTC**), which nowadays draws much attention from the community (Clark et al., 2018; Gururangan et al., 2019; Chen et al., 2020).

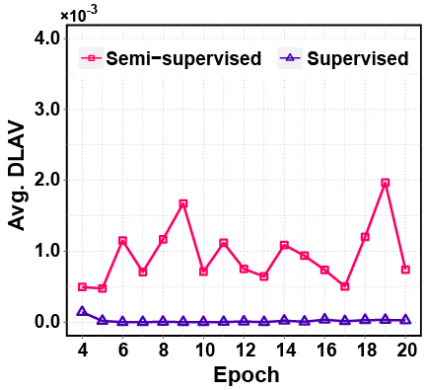

Figure 1: The average difference of label angle variances (Avg.DLAV) computed in semi-supervised and supervised manners across AG News, respectively.

To our knowledge, the most recent SSTC methods mainly borrow ideas from the successful patterns of supervised deep learning, such as pre-training and fine-tuning (Dai and Le, 2015; Howard and Ruder, 2018; Peters et al., 2018; Gururangan et al., 2019; Devlin et al., 2019). Generally, those methods perform deep representation learning on unlabeled texts followed by supervised learning on labeled texts. However, a drawback is that they separately learn from the labeled and unlabeled texts, where, specifically, the deep representations are trained without using the labeling information, resulting in potentially less discriminative representations as well as worse performance.

To avoid this problem, other SSTC methods combine the traditional spirit of self-training with deep learning, which jointly learn the deep representation and classifier using both labeled and unlabeled texts in a unified framework (Miyato et al., 2017, 2019; Sachan et al., 2019; Xie et al., 2020; Chen

[*] Contributing equally with the first author.
[†] Corresponding author.

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, pages 5044–5053
August 1–6, 2021.

et al., 2020). To be specific, this kind of methods initializes a deep classifier, *e.g.,* BERT (Devlin et al., 2019) with Angular Margin (AM) loss (Wang et al., 2018), by training over labeled texts only; and then it alternatively predicts unlabeled texts as their pseudo-labels and trains the deep classifier over the mixture of labeled and pseudo-labeled texts. Accordingly, both labeled and unlabeled texts can directly contribute to the deep classifier training.

Generally speaking, for deep self-training methods, one significant factor of performance is the accuracy of pseudo-labels for unlabeled texts. Unfortunately, they often suffer from low accuracy, where one major reason is the margin bias problem. To interpret this problem, we look around the AM loss with respect to the **label angle**, *i.e.,* the angles between deep representations of texts and weight vectors of labels. For unlabeled texts, the pseudo-labels are predicted by only ranking the label angles, but neglecting the difference between **label angle variances**, *i.e.,* the variance of label angles of texts within the same label, which might be much large in SSL as illustrated in Fig.1. In this context, the boundary of AM loss is actually not the optimal one, potentially resulting in lower accuracy for pseudo-labels (see Fig.2(a)).

To alleviate the aforementioned problem, we propose a novel SSTC method built on BERT with AM loss, namely **S**emi-**S**upervised **T**ext **C**lassification with **B**alanced **D**eep representation **D**istributions (**S^{2}TC-BDD**). Most specifically, in S^{2}TC-BDD, we suppose that the label angles are drawn from each label-specific Gaussian distribution. Therefore, for each text we can apply linear transformation operations to balance the label angle variances. This is equivalent to moving the boundary to the optimal one, so as to eliminate the margin bias (see examples in Fig.2(b)). We can estimate each label angle variance over both labeled and pseudo-labeled texts during the self-training loops. We evaluate the proposed S^{2}TC-BDD method by comparing the most recent deep SSTC methods. Experimental results indicate the superior performance of S^{2}TC-BDD even with very few labeled texts.

2 Related Work

The pre-training and fine-tuning framework has lately shown impressive effectiveness on a variety of tasks (Dai and Le, 2015; Radford et al., 2019a; Howard and Ruder, 2018; Peters et al., 2018; Devlin et al., 2019; Yang et al., 2019; Chen et al., 2019; Akbik et al., 2019; Radford et al., 2019b; Brown et al., 2020; Chen et al., 2020). They mainly perform deep representation learning on generic data, followed by supervised learning for downstream tasks. Several SSTC methods are built on this framework (Dai and Le, 2015; Howard and Ruder, 2018; Peters et al., 2018; Gururangan et al., 2019; Devlin et al., 2019). For instance, the VAriational Methods for Pretraining In Resource-limited Environments (VAMPIRE) (Gururangan et al., 2019) first pre-trains a Variational Auto-Encoder (VAE) model on unlabeled texts, and then trains a classifier on the augmentation representations of labeled texts computed by the pre-trained VAE. However, the VAE model is trained without using the labeling information, resulting in potentially less discriminative representations for labeled texts.

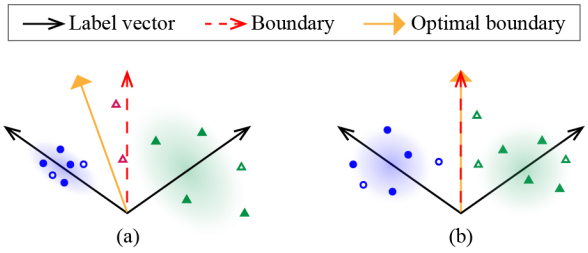

Figure 2: Let solid circle and triangle denote labeled positive and negative texts, and hollow ones denote corresponding unlabeled texts. (a) The large difference between label angle variances results in the margin bias. Many unlabeled texts (in red) can be misclassified. (b) Balancing the label angle variances can eliminate the margin bias. Best viewed in color.

Recent works on SSTC mainly focus on deep self-training (Miyato et al., 2017; Clark et al., 2018; Sachan et al., 2019; Miyato et al., 2019; Xie et al., 2020; Chen et al., 2020), which can jointly learn deep representation and classifier using both labeled and unlabeled texts in a unified framework. It is implemented by performing an alternative process, in which the pseudo-labels of unlabeled texts are updated by the current deep classifier, and then the deep classifier is retrained over both labeled and pseudo-labeled texts. For example, the Virtual Adversarial Training (VAT) method (Miyato et al., 2017, 2019) follows the philosophy of making the classifier robust against random and local perturbation. It first generates the predictions of original texts with the current deep classifier and then trains the deep classifier by utilizing a consistency loss

between the original predictions and the outputs of deep classifier over noise texts by applying local perturbations to the embeddings of original texts. Further, the work in (Sachan et al., 2019) combines maximum likelihood, adversarial training, virtual adversarial training, and entropy minimization in a unified objective. Furthermore, rather than applying local perturbations, Unsupervised Data Augmentation (UDA) (Xie et al., 2020) employs consistency loss between the predictions of unlabeled texts and corresponding augmented texts by data augmentation techniques such as back translations and tf-idf word replacements. The work (Clark et al., 2018) exploits cross-view training by matching the predictions of auxiliary prediction modules over the restricted views of unlabeled texts (*e.g.,* only part of sentence) with ones of primary prediction module over the corresponding full views.

Orthogonal to the aforementioned self-training SSTC methods, our S^{2}TC-BDD further considers the margin bias problem by balancing the label angle variances. This is beneficial for more accurate pseudo-labels for unlabeled texts, so as to boost the performance of SSTC tasks.

3 The Proposed S^{2}TC-BDD Method

In this section, we describe the proposed deep self-training SSTC method, namely **S**emi-**S**upervised **T**ext **C**lassification with **B**alanced **D**eep representation **D**istributions (**S^{2}TC-BDD**).

Formulation of SSTC Consider a training dataset $\mathcal{D}$ consisting of a limited labeled text set $\mathcal{D}_l = \{(\mathbf{x}_i^l, \mathbf{y}_i^l)\}_{i=1}^{i=N_l}$ and a large unlabeled text set $\mathcal{D}_u = \{\mathbf{x}_j^u\}_{j=1}^{j=N_u}$. Specifically, let $\mathbf{x}_i^l$ and $\mathbf{x}_j^u$ denote the word sequences of labeled and unlabeled texts, respectively; and let $\mathbf{y}_i^l \in \{0,1\}^K$ denote the corresponding one-hot label vector of $\mathbf{x}_i^l$, where $y_{ik}^l = 1$ if the text is associated with the k-th label, or $y_{ik}^l = 0$ otherwise. We declare that N_l, N_u, and K denote the numbers of labeled texts, unlabeled texts and category labels, respectively. In this paper, we focus on the paradigm of inductive SSTC, whose goal is to learn a classifier from the training dataset $\mathcal{D}$ with both labeled and unlabeled texts. The important notations are described in Table 1.

Table 1: Summary of notations

Notation	Description
N_l	Number of labeled texts
N_u	Number of unlabeled texts
K	Number of category labels
$\mathcal{D}_l$	Labeled text set
$\mathcal{D}_u$	Unlabeled text set
$\mathbf{x}^l$	Word sequence of labeled text in $\mathcal{D}_l$
$\mathbf{x}^u$	Word sequence of unlabeled text in $\mathcal{D}_u$
$\mathbf{y}^l \in \{0,1\}^K$	One-hot label vector of labeled text

3.1 Overview of S^{2}TC-BDD

Overall speaking, our S^{2}TC-BDD performs a self-training procedure for SSTC. Given a training dataset, it first trains a fine-tuned deep classifier based on the pre-trained BERT model (Devlin et al., 2019) with AM loss (Wang et al., 2018). During the self-training loops, we employ the current deep classifier to predict unlabeled texts as pseudo-labels, and then update it over both labeled and pseudo-labeled texts. In particular, we develop a **B**alanced **D**eep representation **D**istribution (**BDD**) loss, aiming at more accurate pseudo-labels for unlabeled texts. The overall framework of S^{2}TC-BDD is shown in Fig.3. We now present the important details of S^{2}TC-BDD.

BDD Loss Formally, our BDD loss is extended from the AM loss (Wang et al., 2018). For clarity, we first describe the AM loss with respect to angles. Given a training example $(\mathbf{x}_i, \mathbf{y}_i)$, it can be formulated below:

$$\mathfrak{L}_{am}(\mathbf{x}_i, \mathbf{y}_i; \boldsymbol{\phi}) = -\sum_{k=1}^{K} y_{ik} \log \frac{e^{s(\cos(\theta_{ik}) - y_{ik} m)}}{\sum_{j=1}^{K} e^{s(\cos(\theta_{ij}) - y_{ij} m)}}, \quad (1)$$

where $\boldsymbol{\phi}$ denotes the model parameters,

$$\cos(\theta_{ik}) = \frac{\mathbf{f}_i^\top \mathbf{W}_k}{\|\mathbf{f}_i\|_2 \|\mathbf{W}_k\|_2},$$

$\|\cdot\|_2$ is the ℓ_2-norm of vectors; $\mathbf{f}_i$ and $\mathbf{W}_k$ denote the deep representation of text $\mathbf{x}_i$ and the weight vector of label k, respectively; θ_{ik} is the angle between $\mathbf{f}_i$ and $\mathbf{W}_k$; s and m are the parameters used to control the rescaled norm and magnitude of cosine margin, respectively.

Reviewing Eq.1, we observe that it directly measures the loss by label angles of texts only. We kindly argue that it corresponds to non-optimal decision boundary in SSTC, where the difference between label angle variances is much larger than supervised learning. To alleviate this problem, we suppose that the label angles are drawn from each label-specific Gaussian distri-

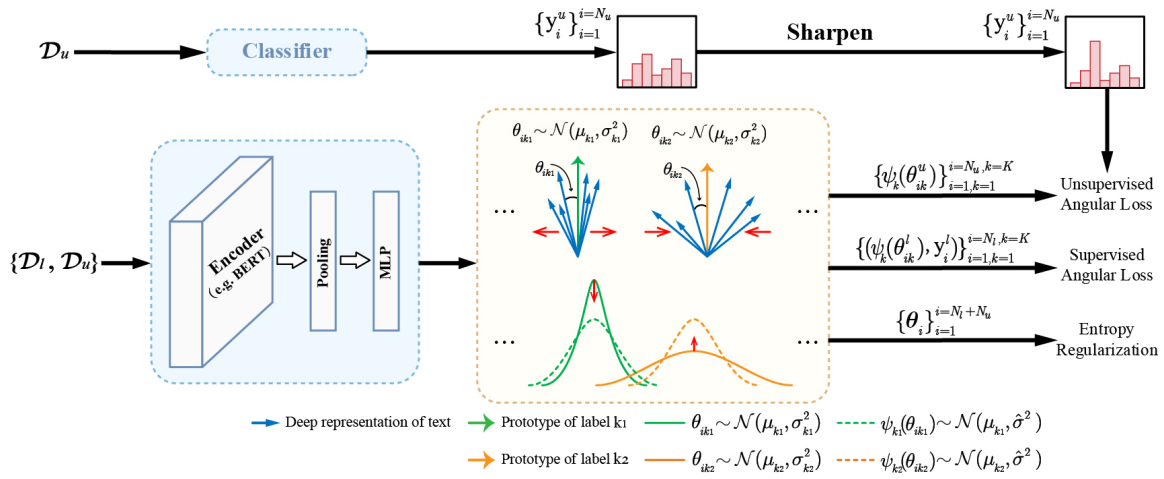

Figure 3: Overview the framework of S^{2}TC-BDD. Best viewed in color.

bution $\{\mathcal{N}(\mu_k, \sigma_k^2)\}_{k=1}^{k=K}$. Thanks to the properties of Gaussian distribution, we can easily transfer them into the ones with balanced variances $\{\mathcal{N}(\mu_k, \widehat{\sigma}^2)\}_{k=1}^{k=K}$, $\widehat{\sigma}^2 = \frac{\sum_{k=1}^{K} \sigma_k^2}{K}$ by performing the following linear transformations to the angles:

$$\psi_k(\theta_{ik}) = a_k \theta_{ik} + b_k, \quad \forall k \in [K], \tag{2}$$

where

$$a_k = \frac{\widehat{\sigma}}{\sigma_k}, \quad b_k = (1 - a_k)\mu_k. \tag{3}$$

With these linear transformations $\{\psi_k(\cdot)\}_{k=1}^{k=K}$, all angles become the samples from balanced angular distributions with the same variances, *e.g.*, $\psi_k(\theta_{ik}) \sim \mathcal{N}(\mu_k, \widehat{\sigma}^2)$. Accordingly, the angular loss of Eq.1 can be rewritten as the following BDD loss:

$$\mathfrak{L}_{bdd}(\mathbf{x}_i, \mathbf{y}_i; \boldsymbol{\phi}) = -\sum_{k=1}^{K} y_{ik} \log \frac{e^{s(\cos(\psi_k(\theta_{ik})) - y_{ik} m)}}{\sum_{j=1}^{K} e^{s(\cos(\psi_j(\theta_{ij})) - y_{ij} m)}}. \tag{4}$$

Supervised Angular Loss Applying the BDD loss $\mathfrak{L}_{bdd}$ of Eq.4 to the labeled text set $\mathcal{D}_l = \{(\mathbf{x}_i^l, \mathbf{y}_i^l)\}_{i=1}^{i=N_l}$, we can formulate the following supervised angular loss:

$$\mathcal{L}_l(\mathcal{D}_l; \boldsymbol{\phi}) = \frac{1}{N_l} \sum_{i=1}^{N_l} \mathfrak{L}_{bdd}(\mathbf{x}_i^l, \mathbf{y}_i^l; \boldsymbol{\phi}). \tag{5}$$

Unsupervised Angular Loss Under the self-training paradigm, we form the loss with unlabeled texts and pseudo-labels. Specifically, we denote the pseudo-label as the output probability of the deep classifier. It is computed by normalizing $\{\cos(\psi_k(\theta_{ik}))\}_{k=1}^{k=K}$ with the softmax function:

$$p(k|\mathbf{x}_i, \boldsymbol{\phi}) = \frac{e^{\cos(\psi_k(\theta_{ik}))}}{\sum_{j=1}^{K} e^{\cos(\psi_j(\theta_{ij}))}} \triangleq \mathbf{y}_i, \ \forall k \in [K].$$

For each unlabeled text $\mathbf{x}_i^u$ the pseudo-label distribution is given by $p(k|\mathbf{x}_i^u, \widetilde{\boldsymbol{\phi}}) \triangleq \mathbf{y}_i^u$ with the *fixed* copy $\widetilde{\boldsymbol{\phi}}$ of the current model parameter $\boldsymbol{\phi}$ during self-training loops. Besides, to avoid those pseudo-label distributions $\{\mathbf{y}_i^u\}_{i=1}^{N_u}$ too uniform, we employ a sharpen function with a temperature T over them:

$$\mathbf{y}_i^u = \text{Sharpen}(\mathbf{y}_i^u, T) = \frac{(\mathbf{y}_i^u)^{1/T}}{\|(\mathbf{y}_i^u)^{1/T}\|_1}, \forall i \in [N_u],$$

where $\|\cdot\|_1$ is the ℓ_1-norm of vectors. When $T \to 0$, the pseudo-label distribution tends to be the one-hot vector.

Applying the BDD loss of Eq.4 to the unlabeled text set $\mathcal{D}_u = \{\mathbf{x}_j^u\}_{j=1}^{j=N_u}$ and pseudo-label distributions $\{\mathbf{y}_i^u\}_{i=1}^{N_u}$, we can formulate the following unsupervised angular loss:

$$\mathcal{L}_u(\mathcal{D}_u, \{\mathbf{y}_i^u\}_{i=1}^{N_u}; \boldsymbol{\phi}) = \frac{1}{N_u} \sum_{i=1}^{N_u} \mathfrak{L}_{bdd}(\mathbf{x}_i^u, \mathbf{y}_i^u; \boldsymbol{\phi}). \tag{6}$$

Entropy Regularization Further, we employ the conditional entropy of $p(y|\mathbf{x}_i, \boldsymbol{\phi})$ as an additional regularization term:

$$\mathcal{R}(\mathcal{D}_l, \mathcal{D}_u; \boldsymbol{\phi}) = -\frac{1}{N_l + N_u} \sum_{\mathbf{x}_i \in \mathcal{D}_l, \mathcal{D}_u} \sum_{k=1}^{K} p(k|\mathbf{x}_i, \boldsymbol{\phi}) \log p(k|\mathbf{x}_i, \boldsymbol{\phi}). \tag{7}$$

This conditional entropy regularization is introduced by (Grandvalet and Bengio, 2004), and also utilized in (Sajjadi et al., 2016; Miyato et al., 2019; Sachan et al., 2019). It also sharpens the output probability of the deep classifier.

Full Objective of S^{2}TC-BDD Combining the supervised angular loss Eq.(5), unsupervised angular loss Eq.(6), and entropy regularization Eq.(7), the full objective of S^{2}TC-BDD can be formulated below:

$$\mathcal{L}(\mathcal{D}_l, \mathcal{D}_u; \boldsymbol{\phi}) = \mathcal{L}_l(\mathcal{D}_l; \boldsymbol{\phi}) + \lambda_1 \mathcal{L}_u(\mathcal{D}_u, \{\mathbf{y}_i^u\}_{i=1}^{N_u}; \boldsymbol{\phi}) + \lambda_2 \mathcal{R}(\mathcal{D}_l, \mathcal{D}_u; \boldsymbol{\phi}), \tag{8}$$

where λ_1 and λ_2 are regularization parameters.

3.2 Implementations of Label Angle Variances

In this section, we describe implementations of label angle variances. As mentioned before, what we concern is the estimations of angular distributions $\{\mathcal{N}(\mu_k, \sigma_k^2)\}_{k=1}^{k=K}$, where their draws are the angles between deep representations of texts and label prototypes denoted by $\{\mathbf{c}_k\}_{k=1}^{k=K}$. Both $\{(\mu_k, \sigma_k^2)\}_{k=1}^{k=K}$ and $\{\mathbf{c}_k\}_{k=1}^{k=K}$ are estimated over both labeled and pseudo-labeled texts during self-training loops. In the following, we describe their learning processes in more detail.

Within the framework of stochastic optimization, we update the $\{(\mu_k, \sigma_k^2)\}_{k=1}^{k=K}$ and $\{\mathbf{c}_k\}_{k=1}^{k=K}$ per-epoch. For convenience, we denote Ω as the index set of labeled and unlabeled texts in one epoch, $\{\mathbf{f}_i\}_{i\in\Omega}$ and $\{\mathbf{y}_i\}_{i\in\Omega}$ as the deep representations of texts and corresponding label or pseudo-label vectors (*i.e.,* $\mathbf{y}_i^l$ or $\mathbf{y}_i^u$) in the current epoch, respectively.

Estimating Label Prototypes Given the current $\{\mathbf{f}_i\}_{i\in\Omega}$ and $\{\mathbf{y}_i\}_{i\in\Omega}$, we calculate the label prototypes $\{\mathbf{c}_k\}_{k=1}^{k=K}$ by the weighted average of $\{\mathbf{f}_i\}_{i\in\Omega}$, formulated below:

$$\mathbf{c}_k = \frac{\sum_{i\in\Omega} y_{ik}\mathbf{f}_i}{\sum_{i\in\Omega} y_{ik}}, \quad \forall k \in [K]. \tag{9}$$

To avoid the misleading affect of some mislabeled texts, inspired by (Liu et al., 2020), we update $\{\mathbf{c}_k\}_{k=1}^{k=K}$ by employing the moving average with a learning rate γ:

$$\mathbf{c}_k^{(t)} \leftarrow (1-\gamma)\mathbf{c}_k^{(t)} + \gamma \mathbf{c}_k^{(t-1)}.$$

Estimating Label Angle Variances Given $\{\mathbf{f}_i\}_{i\in\Omega}$ and $\{\mathbf{c}_k\}_{k=1}^{k=K}$, the angles between them can be calculated by:

$$\beta_{ik} = \arccos\Big(\frac{\mathbf{f}_i^\top \mathbf{c}_k}{\|\mathbf{f}_i\|_2 \|\mathbf{c}_k\|_2}\Big), \quad \forall i \in \Omega, k \in [K]. \tag{10}$$

Accordingly, we can compute the estimations of $\{\mu_k\}_{k=1}^{k=K}$ and $\{\sigma_k^2\}_{k=1}^{k=K}$ as follows:

$$\mu_k = \frac{\sum_{i\in\Omega} y_{ik}\beta_{ik}}{\sum_{i\in\Omega} y_{ik}}, \tag{11}$$

$$\sigma_k^2 = \frac{\sum_{i\in\Omega} y_{ik}(\beta_{ik}-\mu_k)^2}{\sum_{i\in\Omega} y_{ik} - 1}. \tag{12}$$

Further, the moving average is also used to the updates below:

$$\mu_k^{(t)} \leftarrow (1-\gamma)\mu_k^{(t)} + \gamma\mu_k^{(t-1)},$$
$$(\sigma_k^2)^{(t)} \leftarrow (1-\gamma)(\sigma_k^2)^{(t)} + \gamma(\sigma_k^2)^{(t-1)}.$$

4 Experiment

4.1 Experimental Settings

Datasets To conduct the experiments, we employ three widely used benchmark datasets for text classification: *AG News* (Zhang et al., 2015), *Yelp* (Zhang et al., 2015), and *Yahoo* (Chang et al., 2008). For all datasets, we form the unlabeled training set $\mathcal{D}_u$, labeled training set $\mathcal{D}_l$ and development set by randomly drawing from the corresponding original training datasets, and utilize the original test sets for prediction evaluation. The dataset statistics and split information are described in Table 2.

Baseline Models To evaluate the effectiveness of S^{2}TC-BDD, we choose five existing SSTC algorithms for comparison. The details of baseline methods are given below.

- **NB+EM** (Nigam et al., 2000): A semi-supervised text classification method combining a Naive Bayes classifier (NB) and Expectation-Maximization (EM). In experiments, we pre-process texts following (Gururangan et al., 2019) and use tf-idfs as the representations of texts.
- **BERT** (Devlin et al., 2019): A supervised text classification method built on the pre-trained BERT-based-uncased model[1] and fine-tuned with the supervised softmax loss on labeled texts.

[1] https://pypi.org/project/pytorch-transformers/

- **BERT+AM**: A semi-supervised text classification method built on the pre-trained BERT-based-uncased[1] and fine-tuned following the self-training spirit with the AM loss on both labeled and unlabeled texts.

- **VAMPIRE** (Gururangan et al., 2019): A semi-supervised text classification method based on variational pre-training. The code is available on the net.[2] In experiments, the default parameters are utilized.

- **VAT** (Miyato et al., 2019): A semi-supervised text classification method based on virtual adversarial training. [parameter configuration: perturbation size $\epsilon = 5.0$, regularization coefficient $\alpha = 1.0$, hyperparameter for finite difference $\xi = 0.1$]

- **UDA** (Xie et al., 2020): A semi-supervised text classification method based on unsupervised data augmentation with back translation. The code is available on the net.[3] In experiments, we utilize the default parameters, and generate the augmented unlabeled data by using FairSeq[4] with German as the intermediate language.

For S^{2}TC-BDD, BERT, BERT+AM, VAT and UDA, we utilize BERT-based-uncased tokenizer to tokenize texts; average pooling over BERT-based-uncased model as text encoder to encode texts; and a two-layer MLP, whose hidden size and activation function are 128 and tanh respectively, as the classifier to predict labels. We set the max sentence length as 256 and remain the first 256 tokens for texts exceeding the length limit. For optimization, we utilize the Adam optimizer with learning rates of 5e-6 for BERT encoder and 1e-3 for MLP classifier. For BERT, we set the batch size of labeled tests as 8. For S^{2}TC-BDD, BERT+AM, VAT and UDA, the batch sizes of labeled and unlabeled tests are 4 and 8, respectively. For all datasets, we iterate 20 epochs, where each one contains 200 inner loops. All experiments are carried on a Linux server with two NVIDIA GeForce RTX 2080Ti GPUs, Intel Xeon E5-2640 v4 CPU and 64G memory.

Parameter Settings For S^{2}TC-BDD, in our experiments, its parameters are mostly set as: $\lambda_1 = 1.0$, $\lambda_2 = 1.0$, $s = 1.0$, $m = 0.01$. Specifically, for *Yelp* we set $m = 0.3$. For the sharpening temperature T, we set 0.5 for *AG News* and *Yahoo*, 0.3 for *Yelp*. The learning rate γ of label prototypes and label angle variances is set to 0.1.

[2] https://github.com/allenai/vampire
[3] https://github.com/google-research/uda
[4] https://github.com/pytorch/fairseq

Table 2: Statistics of datasets. *#Class*: the number of class labels. *#Labeled*: the number of labeled training texts. *#Unlabeled*: the number of unlabeled training texts. *#Dev*: the number of development texts. *#Test*: the number of texts for testing.

Dataset	#Class	#Labeled	#Unlabeled	#Dev	#Test
AG News	4	10,000	20,000	8,000	7,600
Yelp	5	10,000	20,000	10,000	50,000
Yahoo	10	10,000	40,000	20,000	60,000

Metrics We utilize two metrics of Micro-F1 and Macro-F1, which are two different types of the averaged F1 scores. In experiments, we employ the implementation of Micro-F1 and Macro-F1 in the public Scikit-Learn (Pedregosa et al., 2011) tool.[5]

4.2 Results

For all datasets, we perform each method with five random seeds, and report the average scores.

4.2.1 Varying Number of Labeled Texts

We first evaluate the classification performance of S^{2}TC-BDD with different amounts of labeled texts. For all methods, we conduct the experiments by varying the number of labeled texts N_l over the set $\{100, 1000, 10000\}$ with the number of unlabeled texts $N_u = 20000$ for *AG News* and *Yelp*, and $N_u = 40000$ for *Yahoo*. The classification results of both Micro-F1 and Macro-F1 over all datasets are shown in Table 3, in which the best scores among all comparing baselines are highlighted in boldface. Generally speaking, our proposed S^{2}TC-BDD outperforms the baselines in most cases. Across all datasets and evaluation metrics, S^{2}TC-BDD ranks *1.1* in average. Several observations are made below.

- **Comparing S^{2}TC-BDD against baselines:** First, we can observe that S^{2}TC-BDD consistently dominates the pre-training methods (including BERT and VAMPIRE) on both Micro-F1 and Macro-F1 scores by a big margin, especially when labeled texts are scarce. For example, when $N_l = 100$, the Macro-F1 scores

[5] https://scikit-learn.org/stable/

Table 3: Experimental results of Micro-F1 and Macro-F1 varying the number of labeled texts N_l. The best results are highlighted in boldface.

Metric	Dataset	N_l	NB+EM	BERT	BERT+AM	VAMPIRE	VAT	UDA	S^{2}TC-BDD
Micro-F1	*AG News*	100	0.834	0.839	0.856	0.705	0.868	0.855	**0.872**
		1,000	0.855	0.878	0.879	0.833	0.886	0.883	**0.889**
		10,000	0.874	0.905	0.901	0.876	0.898	0.906	**0.907**
	Yelp	100	0.300	0.344	0.399	0.227	0.244	0.387	**0.417**
		1,000	0.355	0.538	0.544	0.476	0.551	**0.554**	0.552
		10,000	0.404	**0.583**	0.574	0.551	0.566	0.580	**0.583**
	Yahoo	100	0.529	0.564	0.589	0.389	0.534	0.576	**0.618**
		1,000	0.624	0.676	0.679	0.547	0.685	0.672	**0.687**
		10,000	0.659	**0.713**	0.706	0.644	0.701	0.707	**0.713**
Macro-F1	*AG News*	100	0.833	0.840	0.856	0.698	0.867	0.855	**0.872**
		1,000	0.855	0.878	0.879	0.833	0.886	0.883	**0.889**
		10,000	0.873	0.905	0.900	0.876	0.897	0.906	**0.907**
	Yelp	100	0.250	0.324	0.371	0.144	0.197	0.357	**0.403**
		1,000	0.329	0.532	0.535	0.476	0.548	**0.550**	**0.550**
		10,000	0.397	**0.586**	0.562	0.553	0.569	0.576	**0.586**
	Yahoo	100	0.489	0.550	0.573	0.356	0.542	0.567	**0.595**
		1,000	0.616	0.671	0.672	0.545	0.675	0.666	**0.680**
		10,000	0.653	0.708	0.695	0.644	0.697	0.704	**0.709**
Average Rank			6.2	3.6	3.4	6.7	3.8	3.0	**1.1**

of S^{2}TC-BDD are even about 0.17, 0.26 and 0.24 higher than VAMPIRE on the datasets of *AG News*, *Yelp* and *Yahoo*, respectively. Second, when labeled texts are very scarce (*i.e.,* when $N_l = 100$), S^{2}TC-BDD performs better than other self-training baseline methods (*i.e.,* NB+EM, BERT+AM, VAT and UDA) on all datasets, *e.g.,* for Micro-F1 about 0.08 higher than VAT on *Yahoo*. Otherwise, when labeled texts are large, S^{2}TC-BDD can also achieve the competitive performance, even perform better across all datasets.

- **Comparing S^{2}TC-BDD against BERT+AM and BERT:** Our S^{2}TC-BDD method consistently outperforms BERT+AM and BERT across all datasets and metrics. For example, when $N_l = 100$ the Micro-F1 scores of S^{2}TC-BDD beat those of BERT+AM by $0.01 \sim 0.03$ and those of BERT by $0.03 \sim 0.05$ across all datasets. That is because S^{2}TC-BDD employs both labeled and unlabeled texts for training and can predict more accurate pseudo-labels of unlabeled texts than BERT+AM, benefiting for the classifier training. This result is expected since S^{2}TC-BDD performs a Gaussian linear transformation to balance the label angel variances, so as to eliminate the margin bias, leading to more accurate predicted pseudo-labels of unlabeled texts. Besides, these results empirically prove that unlabeled texts are beneficial to the classification performance.

- **Comparing BERT based methods against NB+EM and VAMPIRE:** All BERT based methods (*i.e.,* BERT, BERT+AM, VAT, UDA and S^{2}TC-BDD) consistently dominate baselines based on small models (*i.e.,* NB+EM, VAMPIRE). For example, when $N_l = 10000$, the Micro-F1 and Macro-F1 scores of BERT are about 0.03, 0.18 and 0.05 higher than those of NB+EM on the datasets of *AG News*, *Yelp* and *Yahoo*, respectively. The observation is expected because BERT is a bigger model, hence can extract more discriminative representations of texts than those from the VAE model used in VAMPIRE and tf-idfs used in NB+EM.

4.2.2 Varying Number of Unlabeled Texts

For NB+EM, BERT+AM, VAMPIRE, VAT, UDA and S^{2}TC-BDD, we also perform the experiments with 100 labeled texts and varying the number of unlabeled texts N_u over the set $\{0, 200, 2000, 20000\}$ for *AG News* and *Yelp*, and $\{0, 400, 4000, 40000\}$ for *Yahoo*. Note that VAMPIRE needs unlabeled texts for pre-training, thus we omit the experiments for VAMPIRE with $N_u = 0$. The classification results are reported in Table 4. Roughly, for all methods the classification

Table 4: Experimental results of Micro-F1 and Macro-F1 varying the number of unlabeled texts N_u.

Metric	Dataset	N_u	NB+EM	BERT+AM	VAMPIRE	VAT	UDA	S^{2}TC-BDD
Micro-F1	*AG News*	0	0.668	0.844	–	**0.846**	0.839	0.844
		200	0.696	0.855	0.329	0.850	0.844	**0.857**
		2,000	0.752	0.856	0.421	**0.870**	0.853	0.863
		20,000	0.834	0.856	0.705	0.868	0.855	**0.872**
	Yelp	0	0.317	0.381	–	0.341	0.344	**0.395**
		200	0.307	0.385	0.238	0.299	0.397	**0.403**
		2,000	0.302	0.393	0.211	0.294	0.379	**0.417**
		20,000	0.300	0.399	0.227	0.244	0.387	**0.417**
	Yahoo	0	0.312	0.581	–	0.557	0.564	**0.590**
		400	0.318	0.582	0.162	0.519	0.508	**0.593**
		4,000	0.442	0.584	0.221	0.523	0.559	**0.598**
		40,000	0.529	0.589	0.389	0.534	0.576	**0.618**
Macro-F1	*AG News*	0	0.667	0.843	–	**0.845**	0.840	0.843
		200	0.695	0.855	0.219	0.850	0.843	**0.857**
		2,000	0.751	0.855	0.341	**0.870**	0.852	0.864
		20,000	0.833	0.856	0.698	0.867	0.855	**0.872**
	Yelp	0	0.316	0.368	–	0.256	0.324	**0.385**
		200	0.279	0.370	0.161	0.278	0.344	**0.372**
		2,000	0.286	0.379	0.124	0.287	0.362	**0.380**
		20,000	0.250	0.371	0.144	0.197	0.357	**0.403**
	Yahoo	0	0.303	0.567	–	0.562	0.550	**0.585**
		400	0.301	0.571	0.074	0.521	0.500	**0.586**
		4,000	0.420	0.574	0.175	0.524	0.550	**0.590**
		40,000	0.489	0.573	0.356	0.542	0.567	**0.595**
Average Rank			4.8	2.2	6.0	3.4	3.4	**1.2**

Table 5: Classification performance on *AG News* with 100 labeled data and 20,000 unlabeled data after removing different parts of S^{2}TC-BDD.

Model	Micro-F1	Macro-F1
S^{2}TC-BDD	**0.872**	**0.872**
-entropy regularization	0.863	0.864
-BDD	0.856	0.856
-unlabeled texts	0.844	0.843
-all	0.839	0.840

performance becomes better as the amount of unlabeled texts increasing. For instance, the Micro-F1 scores of S^{2}TC-BDD on all datasets gain about 0.3 improvement as the number of unlabeled texts increasing. These results prove the effectiveness of unlabeled texts in riching the limited supervision from scarce labeled texts and improving the classification performance. Besides, an obvious observation is that the self-training methods (*i.e.,* NB+EM, BERT+AM, VAT, UDA and S^{2}TC-BDD) consistently outperform the pre-training method (*i.e.,* VAMPIRE), especially when unlabeled texts are fewer. The possible reason is that the pre-training methods need more unlabeled texts for pre-training while the self-training methods do not have the requirement.

4.3 Ablation Study

We perform ablation studies by stripping each component each time to examine the effectiveness of each component in S^{2}TC-BDD. Here, we denote BDD as balanced deep representation angular loss $\mathfrak{L}_{bdd}$ in Eq.4. Stripping BDD means that we replace the proposed loss $\mathfrak{L}_{bdd}$ with the AM loss $\mathfrak{L}_{am}$ in Eq.1. The results are displayed in Table 5. Overall, the classification performance will drop when removing any component of S^{2}TC-BDD, suggesting that all parts make contributions to the final performance of S^{2}TC-BDD. Besides, removing unlabeled texts brings the most significant drop of the performance. This result is expected because label angle variances approximated only with very scarce labeled texts will have lower accuracy, resulting in worse performance. Further, in contrast to entropy regularization, the performance after stripping BDD decrease more. Note that the difference between the proposed $\mathfrak{L}_{bdd}$ and $\mathfrak{L}_{am}$ is whether constraining the label angle variances to be balanced or not. This result indicates that the balanced constraint of label angle variances brings a better deep classifier as well as more accurate pseudo-labels for unlabeled texts, especially when labeled texts are limited, and also empirically prove the

Table 6: Average per-epoch running time (second, **s**) of BERT, BERT+AM and S^{2}TC-BDD.

Dataset	**BERT**	**BERT+AM**	**S^{2}TC-BDD**
AG News	72.1 s	71.9 s	73.3 s
Yelp	73.4 s	73.8 s	73.8 s
Yahoo	74.1 s	75.1 s	75.1 s

effectiveness of our balanced label angle variances.

4.4 Efficiency Comparison

To evaluate the efficiency of our S^{2}TC-BDD, we perform efficiency comparisons over BERT, BERT+AM and S^{2}TC-BDD on all benchmark datasets. To be fair, for all methods and datasets we set the batch sizes of labeled and unlabeled texts to 4 and 8 respectively, and iterate 100 epochs, where each one consists of 200 inner loops. The average per-epoch running time results are shown in Table 6. Generally speaking, the per-epoch running time of our proposed S^{2}TC-BDD is close to those of BERT and BERT+AM. This result means that Gaussian linear transformation and estimation of label angle variances in our S^{2}TC-BDD only introduce very few computation costs. That is expected since they merely require very few simple linear operations, which are very efficient.

5 Conclusion

In this paper, we propose a novel self-training SSTC method, namely S^{2}TC-BDD. Our S^{2}TC-BDD addresses the margin bias problem in SSTC by balancing the label angle variances, *i.e.,* the variance of label angles of texts within the same label. We estimate the label angle variances with both labeled and unlabeled texts during the self-training loops. To constrain the label angle variances to be balanced, we design several Gaussian linear transformations and incorporate them into a well established AM loss. Our S^{2}TC-BDD empirically outperforms the existing SSTC baseline methods.

Acknowledgments

We would like to acknowledge support for this project from the National Natural Science Foundation of China (NSFC) (No.61876071, No.62006094), the Key R&D Projects of Science and Technology Department of Jilin Province of China (No.20180201003SF, No.20190701031GH).

References

Alan Akbik, Tanja Bergmann, and Roland Vollgraf. 2019. Pooled contextualized embeddings for named entity recognition. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 724–728.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Ming-Wei Chang, Lev-Arie Ratinov, Dan Roth, and Vivek Srikumar. 2008. Importance of semantic representation: Dataless classification. In *AAAI Conference on Artificial Intelligence*, pages 830–835.

Jiaao Chen, Jianshu Chen, and Zhou Yu. 2019. Incorporating structured commonsense knowledge in story completion. In *AAAI Conference on Artificial Intelligence*, pages 6244–6251.

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

Kevin Clark, Minh-Thang Luong, Christopher D. Manning, and Quoc V. Le. 2018. Semi-supervised sequence modeling with cross-view training. In *Conference on Empirical Methods in Natural Language Processing*, pages 1914–1925.

Andrew M. Dai and Quoc V. Le. 2015. Semi-supervised sequence learning. In *Neural Information Processing Systems*, pages 3079–3087.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186.

Jesper E. van Engelen and Holger H. Hoos. 2020. A survey on semi-supervised learning. *Machine Learning*, 109(2):373–440.

Yves Grandvalet and Yoshua Bengio. 2004. Semi-supervised learning by entropy minimization. In *Neural Information Processing Systems*, pages 529–536.

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

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Annual Meeting of the Association for Computational Linguistics*, pages 328–339.

Jialun Liu, Yifan Sun, Chuchu Han, Zhaopeng Dou, and Wenhui Li. 2020. Deep representation learning on long-tailed data: A learnable embedding augmentation perspective. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2970–2979.

Takeru Miyato, Andrew M. Dai, and Ian J. Goodfellow. 2017. Adversarial training methods for semi-supervised text classification. In *International Conference on Learning Representations*.

Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. 2019. Virtual adversarial training: A regularization method for supervised and semi-supervised learning. *IEEE Transactions Pattern Analysis and Machine Intelligence*, 41(8):1979–1993.

Kamal Nigam, Andrew McCallum, Sebastian Thrun, and Tom M. Mitchell. 2000. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39(2):103–134.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake VanderPlas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Edouard Duchesnay. 2011. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2227–2237.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2019a. Improving language understanding by generative pre-training.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019b. Language models are unsupervised multitask learners.

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

Mehdi Sajjadi, Mehran Javanmardi, and Tolga Tasdizen. 2016. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. In *Neural Information Processing Systems*, pages 1171–1179.

Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu. 2018. Cosface: Large margin cosine loss for deep face recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5265–5274.

Qizhe Xie, Zihang Dai, Eduard H. Hovy, Thang Luong, and Quoc Le. 2020. Unsupervised data augmentation for consistency training. In *Neural Information Processing Systems*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Neural Information Processing Systems*, pages 5753–5763.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Neural Information Processing Systems*, pages 649–657.