Contrast-Enhanced Semi-supervised Text Classification with Few Labels

Austin Cheng-Yun Tsai, Sheng-Ya Lin, Li-Chen Fu

Department of Computer Science and Information Engineering, National Taiwan University
{r08922086, r09944044, lichen}@ntu.edu.tw

Abstract

Traditional text classification requires thousands of annotated data or an additional Neural Machine Translation (NMT) system, which are expensive to obtain in real applications. This paper presents a Contrast-Enhanced Semi-supervised Text Classification (CEST) framework under label-limited settings without incorporating any NMT systems. We propose a certainty-driven sample selection method and a contrast-enhanced similarity graph to utilize data more efficiently in self-training, alleviating the annotation-starving problem. The graph imposes a smoothness constraint on the unlabeled data to improve the coherence and the accuracy of pseudo-labels. Moreover, CEST formulates the training as a "learning from noisy labels" problem and performs the optimization accordingly. A salient feature of this formulation is the explicit suppression of the severe error propagation problem in conventional semi-supervised learning. With solely 30 labeled data per class for both training and validation dataset, CEST outperforms the previous state-of-the-art algorithms by 2.11% accuracy and only falls within the 3.04% accuracy range of fully-supervised pre-training language model fine-tuning on thousands of labeled data.

Introduction

Text classification is one of the most fundamental tasks in the Natural Language Processing (NLP) research community, with a broad range of applications, such as question answering, sentiment analysis, topic mining, and spam detection. Previous researches have developed several deep-learning-based algorithms (Kim 2014; Zhang, Zhao, and LeCun 2015; Tang, Qin, and Liu 2015; Yang et al. 2016) and have achieved great success when abundant labeled data are provided (usually over tens of thousands). However, when there is only a limited number of labeled data, complex neural networks often suffer from over-fitting (Xie et al. 2020). As a result, increasing attention has been paid to semi-supervised learning (SSL) to effectively utilize large amounts of unlabeled data to address the above shortcomings since unlabeled data are much easier and cheaper to collect (Chawla and Karakoulas 2005).

Recent semi-supervised approaches for text classification primarily focus on exploiting the consistency in the predictions for the same samples under different perturbations (Miyato et al. 2018, 2017; Xie et al. 2020; Chen, Yang, and Yang 2020). However, they typically involve a sophisticated Neural Machine Translation (NMT) system for augmenting data by back-translation, which translates a sentence into a different language and then translates it back. Such approaches may be bothersome in real-world scenarios by requiring an additional NMT system. Moreover, the system may generate poor-quality sentences if the task-specific data distribution is different from that of the data on which the NMT system was pre-trained.

One promising solution to this issue is self-training (Lee et al. 2013; Grandvalet and Bengio 2004; Meng et al. 2020). Self-training generates pseudo-labels for unlabeled data, which were later used as new labeled data for further training. Traditional self-training does not perform sample selection, nor does it consider noises in the generated pseudo-labels during training. This may result in error accumulation (Zhang et al. 2017; Arazo et al. 2020) throughout training iterations, which is referred to as confirmation bias in contemporary works (Tarvainen and Valpola 2017; Arazo et al. 2020) and is considered a severe problem in conventional SSL approaches. Though prior works have been devoted to designing criteria for selecting samples, for example, selecting samples with smaller loss or higher confidence scores in neural network's predictions, these selection strategies still suffer from over-fitting on the samples that the model is already certain about or from learning on wrong labels.

In this paper, we propose *Contrast-Enhanced Semi-supervised Text Classification (CEST)* framework to overcome all the drawbacks mentioned above. The framework overview is illustrated in Fig. 1. Based on self-training, *CEST* leverages Bayesian Neural Network (BNN) (Wang and Yeung 2016; Gal and Ghahramani 2016) to provide certainty estimates for unlabeled data and judiciously select appropriate instances to improve the self-training process. We then build a reliable similarity graph on the selected data instances to enhance the contrast and exploit the smoothness assumption in SSL among data instances. Inducing the contrast between data not only encourages the model to learn better representations but also results in better generality and better efficiency in utilizing data. Finally, we re-formulate the training on pseudo-labels as a new problem and learn in a noise-robust manner to alleviate the severe confirmation bias issue.

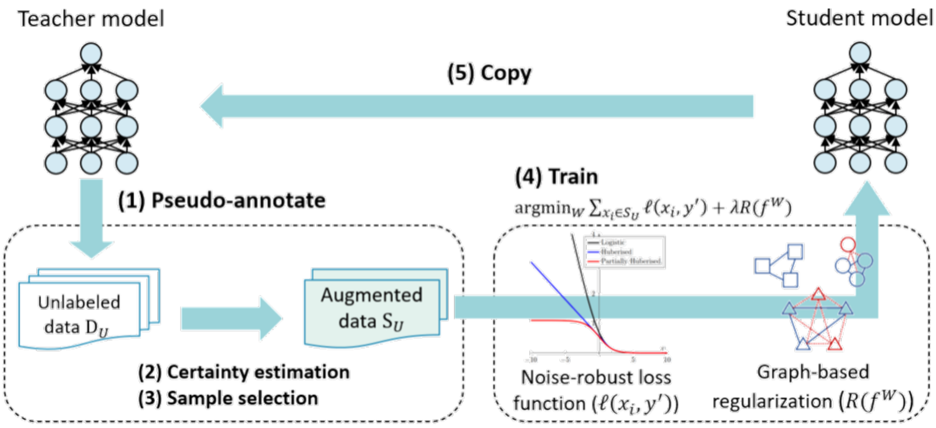

Figure 1: Framework overview.

We perform large-scale experiments on five benchmark datasets with merely 30 labeled data for training and validation datasets per class. We adopt BERT as our base encoder and show that our algorithm significantly improves BERT's average performance by 7.96%. Moreover, on benchmark datasets, *CEST* significantly outperforms all the previous state-of-the-art algorithms, most of which require supplementary resources such as additional data augmentation modules or a sophisticated NMT system. In summary, our proposed framework makes the following contributions: (i) We propose a semi-supervised text classification framework *CEST* under label-limited settings without using any auxiliary resources. (ii) We propose reliable similarity graph to empower self-training to consider smoothness in SSL, which conventional self-training does not. (iii) We propose a new problem formulation for self-training to mitigate the severe confirmation bias in conventional SSL. (iv) We show substantial performance improvement of *CEST* over the state-of-the-art algorithms on benchmark datasets by 2.11% in accuracy with an overall 91.57% accuracy, only 3.04% accuracy short to fully-supervised learning that uses at least 830 times more labeled data.

Related Work

Semi-supervised text classification has been extensively studied in the NLP community. Gururangan et al. (2019), Chen et al. (2018), and Yang et al. (2017) established variational auto-encoders-based algorithms that learned to reconstruct sentences and utilized the latent variables to classify text or label sequences. Virtual adversarial training (Miyato, Dai, and Goodfellow 2017) perturbed word embeddings to encourage consistency between perturbed embeddings. Unsupervised data augmentation (UDA) (Xie et al. 2020) performed consistency training by making features consistent between back-translated sentences (Sennrich, Haddow, and Birch 2016). MixText (Chen, Yang, and Yang 2020) created virtual training samples by interpolating in BERT's hidden states and performed similar consistency training as UDA. Most of the above methods require additional modules to facilitate the main module's training, which is inefficient in practical applications and may suffer from the mismatch in data distribution between the additional modules and the target tasks. Uncertainty-aware Self-Training (UST) (Mukherjee and Awadallah 2020) utilized BNN to select samples and performed self-training on the selected data. However, it was still afflicted by the confirmation bias issue and did not consider smoothness among data in the feature space.

Preliminaries

We first define the notations used in this paper. Assume that we are given a labeled dataset $D_L = \{x_i, y_i\}_{i=1}^{N_L}$ and an unlabeled dataset $D_U = \{x_i\}_{i=1}^{N_U}$, where $x_i \in \mathbb{X}$ is a text sequence, and $y_i \in \mathbb{Y}$ is the corresponding label. N_L and N_U are the number of labeled and unlabeled data, respectively ($N_L \ll N_U$). The goal of semi-supervised text classification is to learn a mapping function $f^W : \mathbb{X} \to \mathbb{Y}$, where W is the model parameters, by minimizing the empirical loss:

$$\underset{W}{\text{argmin}} \sum_{x_i \in D_L \cup D_U} \ell(x_i, y') + \lambda R(f^W) \qquad (1)$$

where y' is the target label. If x_i is from D_U, y' is defined as the predicted label (pseudo-label). $\ell(\cdot, \cdot) : \mathbb{Y} \times \mathbb{Y} \to \mathbb{R}$ is the loss function measuring the classification loss between the predictions and the labels, typically the cross-entropy loss. R is the regularization term that prevents the model from overly aggressively learning from data. λ is the hyperparameter controlling the impact of R.

Smoothness Assumption in SSL

Semi-supervised learning typically assumes that the learning algorithm should follow the "smoothness assumption." The smoothness assumption states that if two points x_1, x_2 are close in a high-density region, then so should be the corresponding labels y_1, y_2. If they are in a low-density region, their labels should be different, as illustrated in Fig. 1.

Self-training

Self-training first trains a base *teacher* model on the labeled set D_L. The *teacher* model is then used to pseudo-annotate on the deliberately selected $S_U \subseteq D_U$ to form the augmented data, which will be used to train the *student* model. The selection process of S_U should be carefully designed for better performance, for example, selecting data with higher confidence scores, which implies that the data are mostly correctly labeled. Then, the *teacher* model is updated by the *student* model. The teacher-student training process is repeated until convergence. With self-training mechanism, we modify the empirical loss as

$$\underset{W}{\operatorname{argmin}} \sum_{x_i \in S_U, S_U \subseteq D_U} \ell(x_i, \widetilde{y_i}) + \lambda R(W) \tag{2}$$

where W is the model parameters for the *student* model, and the hard pseudo-labels $\widetilde{y_i}$ are given by the *teacher* model W^* from the last iteration (W^* will be fixed in current iteration):

$$\widetilde{y_i} = \underset{c}{\operatorname{argmax}}\, p(y = c|f^{W^*(x)}) \tag{3}$$

Similar design of using hard pseudo-labels instead of soft pseudo-labels has also been reported in contemporary works (Kumar, Ma, and Liang 2020; Chen, Yang, and Yang 2020), which refer it to as *label sharpening*.

Bayesian Neural Network (BNN)

Instead of having deterministic weights, BNN assumes a prior distribution over its model parameters. Considering the mapping function f^W for BNN, where W is the model parameters, the parameter optimization is achieved by finding the posterior distribution over model parameters $p(W|D_{train})$ on a training dataset D_{train}. During inference, for data instance x, the probability for class c is $p(y = c|x) = \int_W p(y = c|f^W(x))p(W|D_{train})dW$. However, it is computationally intractable to calculate over all possible W and we have to find a surrogate distribution $q_\theta(W)$ in a tractable family of distributions to replace the true model posterior $p(W|D_{train})$. Gal and Ghahramani (2016) and Gal, Islam, and Ghahramani (2017) developed Monte-Carlo Dropout (MC Dropout) using BNN and showed that the probability for class c, $p(y = c|x)$, can be approximated by considering $q_\theta(W)$ to be the dropout distribution (Srivastava et al. 2014), which is tractable, with T masked model weights $\{\widetilde{W_t}\}_{t=1}^T \sim q_\theta(W)$:

$$p(y = c|x, D_{train}) \approx \frac{1}{T}\sum_{t=1}^{T} p(y = c|f^{\widetilde{W_t}}(x)) \tag{4}$$

Methodology

The overall framework is illustrated in Fig. 1. In this section, we will answer the following questions: (1) How to select appropriate unlabeled samples for self-training? (2) How to induce contrast among data samples by designing the regularization term $R(\cdot)$? (3) How to robustly learn with pseudo-labeled samples by the classification loss $\ell(\cdot,\cdot)$?

Sample Selection

Certainty estimates. We will select the unlabeled data instances for self-training by first estimating their certainties, as we want to judiciously select samples to prevent the model from being contaminated by wrong pseudo-labels during self-training. To this end, we leverage information gain of model parameters as the certainty measure to estimate how certain the model is to the given sample with respect to the sample's true label, even though the real label is unknown. Similar techniques are also used in (Houlsby et al. 2011; Gal, Islam, and Ghahramani 2017), where they select data with higher information gain for active learning.

Using entropy $\mathbb{H}(\cdot)$ to measure the level of information we have, we define the information gain $\mathbb{B}$ to be the difference between the final entropy $\mathbb{H}(y|x, D_U)$ after seeing the whole unlabeled dataset D_U and the current entropy $\mathbb{H}(y|x, W)$ given the model parameters W in the current iteration. Formally, for data sample $x \in D_U$, the information gain $\mathbb{B}$ with respect to its expected label is

$$\mathbb{B}(y, W|x, D_U) = \mathbb{H}(y|x, D_U) - \mathbb{E}_{p(W|D_U)}[\mathbb{H}(y|x, W)] \tag{5}$$

where $p(W|D_U)$ is the posterior distribution of the model parameter in current iteration. As Eq. (5) is computationally intractable, it can be approximated by MC Dropout (Gal, Islam, and Ghahramani 2017):

$$\begin{aligned}\widehat{\mathbb{B}}(y, W|x, D_U) = &-\sum_c \Big(\frac{1}{T}\sum_t \hat{p}_c^t\Big) \log \Big(\frac{1}{T}\sum_t \hat{p}_c^t\Big) \\ &+ \frac{1}{T}\sum_t \sum_c \hat{p}_c^t \log\,(\hat{p}_c^t)\end{aligned} \tag{6}$$

where $\hat{p}_c^t = p(y = c|f^{\widetilde{W_t}(x)})$ is the estimated probability of class c given by the model parameters $\widetilde{W_t} \sim q_\theta(W)$ in the t-th trial in the MC Dropout. $\hat{\mathbb{B}}$ is a tractable estimation to $\mathbb{B}$ when T is sufficiently large. Learning from $\mathbb{B}$ decreases the expected posterior entropy in the output space Y and encourages the model to produce low-entropy outputs. A lower $\hat{\mathbb{B}}$ value means that the model is more certain about the sample as there is little to be gained even after seeing the whole unlabeled dataset D_U.

Certainty-driven sample selection With the information gain measure, we can use them to design the sample selection approach. For data with lower information gain, the model is certain about them. However, due to having low information gain, the model learns little from the data. Directly training on the data with low information gain will make the model rapidly over-fit on them. On the other hand, data with higher information gain can contribute more to the learning of the model, but they are also more prone to have wrong pseudo-labels. To balance these two scenarios, we sample data instances with different sampling weights, with lower $\hat{\mathbb{B}}$ instances being sampled more and higher $\hat{\mathbb{B}}$ instances being sampled less. We define the sampling weight to be proportional to its certainty value that is measured by $1 - \hat{\mathbb{B}}$ (higher certainty corresponds to lower information gain). Formally, for data $x_i \in D_U$, the sampling weight s_i for x_i is

$$s_i = \frac{1 - \hat{\mathbb{B}}(y_i, W|x_i, D_U)}{\sum_{x_j \in D_U}[1 - \hat{\mathbb{B}}(y_j, W|x_j, D_U)]} \tag{7}$$

where $\sum_{x_j \in D_U}[1 - \hat{\mathbb{B}}(y_j, W|x_j, D_U)]$ is a normalizing factor. It is worth noting that $1 - \hat{\mathbb{B}}(y_i, W|x_i, D_U)$ is always positive provided that the base in log is equal to the number of classes C. To select unlabeled samples for augmented dataset S_U, we sample P instances from D_U, where each data sample will not be sampled twice to avoid over-fitting.

Graph-based Contrast Induction

To consider the smoothness assumption in SSL, We proposed a dynamic similarity graph to induce the contrast among data samples. Traditionally, only perturbation-based SSL approaches consider the smoothness assumption, while self-training-based approaches do not. The proposed dynamic similarity graph empowers self-training to exploit the smoothness and to leverage more underlying information among data, instead of solely considering their pseudo-labels as in traditional self-training, which greatly increases the efficiency of data utilization.

Specifically, we build a similarity graph based on the pseudo-labels and create a reliable sub-graph by pruning the "unreliable" edges. Then, we optimize the model under the guidance of the reliable similarity sub-graph. The optimization explicitly enhances the contrast between data samples and encourages the smoothness in the feature space.

Similarity-graph Construction. After the unlabeled data samples are selected to form S_U, we build an similarity graph on top of S_U. Due to the discrete nature of text, it is hard to measure the similarity between two text sequences. We address this problem in the label space $\mathbb{Y}$ and regard data instances from the same class to be similar. Formally, we build an undirected, weighted similarity graph $G(V, E)$, where the node set V and the edge set E is defined as

$$\begin{aligned} V &= \{i \mid x_i \in S_U\} \\ E &= \{(i,j) \mid x_i, x_j \in S_U\} \end{aligned} \tag{8}$$

and edge (i, j) has edge weight satisfying the following:

$$e_{ij} = \begin{cases} 1, & \text{if } \tilde{y}_i = \tilde{y}_j \\ 0, & \text{if } \tilde{y}_i \neq \tilde{y}_j \end{cases} \tag{9}$$

where $\widetilde{y}_i, \widetilde{y}_j$ are the hard pseudo-labels of x_i obtained by the teacher model W^* from the last iteration.

Graph-based Contrast Enhancement. After the similarity graph is constructed, we show how to enhance the contrast by the graph during training. We first decompose the mapping function f into two components, a feature extractor h and a classifier g with $f = g \circ h$, where h projects $x \in \mathbb{X}$ into its high-level representation $h(x)$ in the feature space $\mathbf{Z}$, and g annotates the representation $h(x)$ with a label $y \in \mathbb{Y}$.

An ideal feature extractor h separates and disentangles $h(x)$ from different classes in $\mathbf{Z}$, and a simple classifier g suffices (Luo et al. 2018). To achieve this goal, we propose to force similar data to have consistent features and make dissimilar data farther away from each other in $\mathbf{Z}$. Mathematically, given a similarity graph G and a pre-defined margin m between dissimilar features, we enhance the contrast through the regularization term R in the training objective:

$$\begin{aligned} R(f^W, G) = &\sum_{(i,j)\in E, e_{ij}=1} ||h(x_i) - h(x_j)||_2^2 \\ &+ \sum_{(i,j)\in E, e_{ij}=0} \max(0, m - ||h(x_i) - h(x_j)||_2)^2 \end{aligned} \tag{10}$$

The loss R is vital to the data efficiency as it mines the underlying information from data, in lieu of using the text data separately without considering their relationships within a class and between classes. Furthermore, from the perspective of SSL, it explicitly forces the model to follow the smoothness assumption (Sec.), making features of the same class be in a high-density region and features of different classes be separated by low-density regions. A smoother decision boundary can thus be obtained.

Reliable Sub-graph Construction. It will be problematic to consider all the edges in the similarity graph and optimize directly since many edges may have wrong weights due to wrong pseudo-labels. Moreover, if we directly optimize the model using the full similarity graph, the regularization term R will in turn confuse the model since it tells the model to put points that should have been in different classes closer.

To this end, we add an attribute, reliability, to each node and each edge in the similarity graph G to assess its quality, and the edges with lower reliability values are pruned to avoid adverse effects caused by those low-quality edges. For node v_i, which corresponds to the data instance x_i, we define its reliability $\gamma(v_i)$ to be the predictive variance of its pseudo-label over T MC Dropout iterations. Specifically,

$$\gamma(v_i) = 1 - \texttt{Var}(y = \tilde{y}_i|x_i) \approx 1 - \frac{1}{T}\sum_{t=1}^{T} p_t^2 + (\frac{1}{T}\sum_{t=1}^{T} p_t)^2 \tag{11}$$

where $p_t = p(y = \tilde{y}_i|f^{\widetilde{W_t}}(x_i)$ is the probability of $\tilde{y}_i$ predicted by $f^{\widetilde{W_t}}$, where $\widetilde{W_t}$ is the model weight sampled in the t-th iteration in MC Dropout. It is noteworthy that the variance $\texttt{Var}(y|x_i)$ falls in the interval $[0, \frac{1}{4}]$, so the value of reliability $\gamma(v_i) = 1 - \texttt{Var}(y|x_i)$ is always positive and falls in $[\frac{3}{4}, 1]$. Then, the reliability of edge (i, j) is

$$\gamma(e_{ij}) = \frac{\gamma(v_i) + \gamma(v_j)}{2} \tag{12}$$

An edge with a higher reliability value indicates that the adjacent nodes are more likely to have correct pseudo-labels, and hence the edge weight is correctly assigned.

To construct the reliable sub-graph $G'(V', E')$, we add all the nodes in G into G', *i.e.* $V' = V$. As for the edge set E', for each node v_i, we select the top k edges with the highest reliability γ from the positive set and from the negative set to form the reliable edge set E' ($2k$ edges in total for each node), where k is a hyper-parameter, and the positive set and the negative set are defined as

1. positive set: $\{(i, j) \mid e_{ij} = 1, j = 1, 2, ..., |S_U|\}$
2. negative set: $\{(i, j) \mid e_{ij} = 0, j = 1, 2, ..., |S_U|\}$

In this way, we have a reliable graph with much fewer wrong edges, and we can still perform the same optimization on the resulting sub-graph G' as before:

$$R(f^W, G') = \sum_{(i,j)\in E', e_{ij}=1} ||h(x_i) - h(x_j)||_2^2 + \sum_{(i,j)\in E', e_{ij}=0} \max(0, m - ||h(x_i) - h(x_j)||_2)^2 \quad (13)$$

For each node, the time complexity of calculating the loss incurred by adjacent edges is significantly reduced from $O(n)$ to $O(k)$, where $k \ll n$. In section , we show that the reliable similarity sub-graph greatly improves the performance, while adding little computational cost.

Robust Learning with Pseudo-labels

Erroneous pseudo-labels are inevitable in self-training, though we have designed several techniques to avoid them. For example, it is still possible that the pseudo-labels of low $\hat{\mathbb{B}}$ data are wrong. The inclusion of higher $\hat{\mathbb{B}}$ instances introduces even more wrongly-annotated data, further worsening the situation. These wrongly-annotated data mislead the learning of the model, and errors accumulate throughout training, degrading the performance.

Utilizing traditional cross-entropy loss to train pseudo-annotated data is prone to error accumulation. Consider $l(f^W(x), y) = \varphi(p_W(x, y))$ to be the loss function, where $p_W(x, y)$ is the probability of the target class y predicted by the model W, and $\varphi(u)$ is the classification loss. For cross-entropy loss, $l(f^W(x), y) = -\ln(p_W(x, y))$. Then, the gradient induced by the cross-entropy loss is

$$\frac{\partial l(f^W(x), y)}{\partial W} = -\frac{\partial \ln p_W(x, y)}{\partial W} = -\frac{1}{p_W(x, y)} \frac{\partial p_W(x, y)}{\partial W} \quad (14)$$

As shown in Eq. (14), the cross-entropy loss implicitly weighs more on *difficult* samples whose predictions are less congruent with the target labels during optimization for faster convergence (Zhang and Sabuncu 2018). However, when labels are noisy, the data with wrong labels tend to be more *difficult* than those having clean labels. The model will aggressively memorize the wrong information through optimization, finally leading to the confirmation bias issue.

To this end, we frame our training on the selected unlabeled data as a *learning from noisy labels* problem to alleviate this issue. We leverage the recent advanced noise-robust loss, partially huberised Loss, (Menon et al. 2020) field, for training. Specifically, by only clipping the gradient of the classification loss φ in the general loss $l(f^W(x), y)$, the resulting partially huberised loss $\widetilde{l}(f^W(x), y)$ is robust to label noises. Mathematically, $\widetilde{l}(f^W(x), y)$ is

$$\widetilde{l} = \begin{cases} -\tau p_W(x, y) - \varphi^*(-\tau), & \text{if } \varphi'(p_W(x, y)) \leq -\tau \\ -\log p_W(x, y), & \text{otherwise} \end{cases} \quad (15)$$

Integrating it with cross-entropy loss , the partially huberised cross-entropy loss (PHCE loss) is obtained:

$$\widetilde{l} = \begin{cases} -\tau p_W(x, y) + \log \tau + 1, & \text{if } p_W(x, y) \leq \frac{1}{\tau} \\ -\log p_W(x, y), & \text{otherwise} \end{cases} \quad (16)$$

Dataset	Class	Train	Test	Unlabeled
DBpedia	14	560K	50K	-
AG News	4	120K	7600	-
IMDB	2	25K	25K	50K
Elec	2	25K	25K	-
SST-2	2	68.2K	1821	-

Table 1: Dataset statistics.

where τ is a hyper-parameter related to the degree of noise. τ is set larger if the data are essentially noise-free. We will use the PHCE loss as our classification loss ℓ in Eq. (1). Menon et al. (2020) theoretically showed that, using the partially harburised loss, the performance degradation under label corruption can be bounded, thus ensuring robustness. The intuition behind this is not to overly trust any single data instance and linearize the loss if the samples are too *difficult*. On the other hand, in the *learning from noisy labels* problem, the model typically suffers from having class-imbalanced data, as the learning will be biased. Hence, we separate the unlabeled data into different classes according to their pseudo-labels, select an equal number of data from each class, and train the model in a class-balanced fashion.

Experiment

Dataset and Evaluation Setting

We evaluate *CEST* on five public datasets (Table 1), including IMDB (Maas et al. 2011), SST-2 (Socher et al. 2013), Elec (McAuley and Leskovec 2013) for sentiment analysis and DBpedia (Mendes, Jakob, and Bizer 2012), AG News (Zhang, Zhao, and LeCun 2015) for topic classification. We randomly select 30 labeled data per class with different random seeds for training and validation set and use the test set in original dataset. The rest data are added to the unlabeled set. We perform each experiment three times with different seeds and data splits to show the significance (Dror et al. 2018) and report the mean accuracy on the test set.

Baselines

For fairness, we only compare our results with the semi-supervised learning methods that uses BERT as the base model. The first baseline is BERT (Devlin et al. 2019) directly fine-tuned on the small labeled training set without using unlabeled data. The next baseline is UDA (Xie et al. 2020), which performs consistency training as regularization through back-translation by an additional NMT system. Our third baseline is the standard self-training without considering sample selection, feature relationships, and confirmation bias. Finally, the fourth baseline is UST (Mukherjee and Awadallah 2020), which selects samples by information gain and utilizes cross-entropy loss to perform self-training. For model implementation, we use huggingface's BERT (`bert-base-uncased`) with a two-layered MLP on top of it. We set the maximum token length in sentences to 256 and clip tokens exceeding the limit. The learning rate is fixed to $1e-5$, and hyper-parameters are set to $k = 2, \tau = 10, \lambda = 0.75, |S_U| = 2000, dim(\mathbb{Z}) = 128$.

Dataset	Full training	30 labeled training data per class				
	BERT	**BERT**	**UDA**	**Standard ST**	**UST**	**CEST (Ours)**
AG News	92.98	79.84	85.92	84.07	86.90	**87.05**
DBpedia	99.13	98.01	96.88	97.25	98.30	**98.61**
IMDB	91.26	80.90	89.30	83.81	84.06	**90.20**
Elec	96.48	85.07	89.64	89.50	89.97	**92.26**
SST-2	93.19	74.23	83.58	84.81	88.09	**89.71**
Average	94.61	83.61	89.06	87.89	89.46	**91.57**
		(+0.00%)	(+5.45%)	(+4.28%)	(+5.85%)	(+7.96%)

Table 2: Performance (test accuracy(%)) comparison with baselines. The results are averaged for three runs, with each run taking 3-8 hours on an NVIDIA RTX3090. BERT on the second column means directly fine-tuning without using unlabeled data. Standard ST denotes standard self-training. We reproduced all baselines with PyTorch except that UDA's results are cited.

Results

Table 2 shows the overall performance on the benchmark datasets. With only 30 labels per class in the training set, our model consistently achieves the best performance over all the other baselines with an aggregated accuracy of 91.57%. BERT without using unlabeled data performs the worst as expected since the number of labeled data is extremely insufficient to train BERT's substantial amount of parameters. UDA has greatly improved BERT by 5.45% accuracy, owing to the use of an additional NMT system to perform consistency training. While standard self-training has shown a 4.28% accuracy improvement, it fails to tackle the severe confirmation bias issue, hence resulting in lower performance. UST is a stronger baseline that performs on par with UDA. Still, as UST does not suppress confirmation bias and does not consider the smoothness constraint in SSL, its performance can be considered sub-optimal in contrast with ours. Overall, CEST shows high performance over state-of-the-art approaches, UST, by 2.11% accuracy while performing only 3.04% accuracy short to fully-supervised learning.

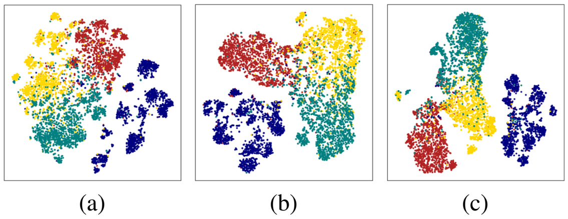

Figure 2: t-SNE visualization. (a) without similarity graph (b) similarity graph without considering reliability (c) with reliable similarity graph.

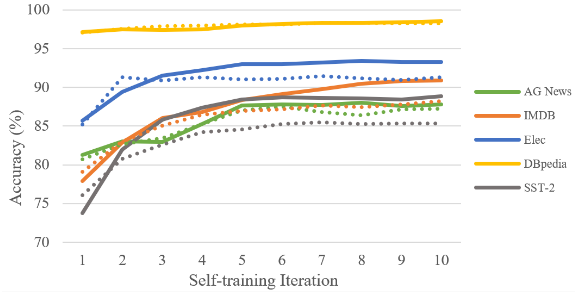

Figure 3: Test accuracy (%) over self-training iterations. The solid and the dotted lines shows the results of training with and without using the noise-robust loss (PHCE loss).

Effectiveness of Reliable Similarity Graph

In Fig. 2, we use t-SNE (Maaten and Hinton 2008) to visualize the learned features in the feature space $\mathbb{Z}$ on the AG News test set. The model trained with reliable similarity graph (Fig. 2c) learned tighter features within each class and separated features between classes. We can see that there is a clearer boundary between every two classes compared to that trained without similarity graph (Fig. 2a), hence restraining over-fitting and yielding better generalization capability. On the other hand, when we use the whole similarity graph without considering the reliability, referring to Fig. 2b, the boundaries are still entangled together because the reliability of edges is not considered in this case, and the wrong features are falsely put closer, and vice versa.

Effectiveness of Noise-robust Loss

In Fig. 3, we demonstrate the importance of using the noise-robust loss function. The performance of training with the PHuberCE loss (the solid lines) consistently has better accuracy against that of training without the noise-robust loss (the dotted lines). Both cases have similar accuracy in the early stages but gradually diverge as the training goes on, suggesting the degradation by the accumulated errors during training iterations and demonstrating the importance of using the noise-robust loss.

Ablation Analysis

In Table 3, we compare the impact of different components in our framework. We remove a component each time to assess its effectiveness. We observe that the performance is the worst after removing the sample selection part since noisy labels will corrupt the training. For the same reason, when we remove the noisy-robust loss, the accuracy drops 0.53%, corroborating our conclusions in section that avoiding noisy labels is imperative to self-training. Moreover, the use of reliable sub-graph and smoothness regularization both

	AG News	**IMDB**	**DBpedia**	**Elec**	**SST-2**	**Average**
BERT (direct fine-tuning)	79.84	80.90	98.01	85.07	74.23	83.61
CEST	**87.05**	**90.20**	**98.61**	**92.26**	**89.71**	**91.57**
− noise-robust loss	86.80	89.68	97.91	92.08	88.75	91.04
− reliable sub-graph	86.56	88.37	97.44	91.96	87.95	90.46
− smoothness regularization	86.10	86.76	98.18	90.47	88.33	89.97
− without sampling	84.07	83.81	97.25	89.51	84.81	87.89

Table 3: Ablation Study of performance (test accuracy (%)) over different design configurations.

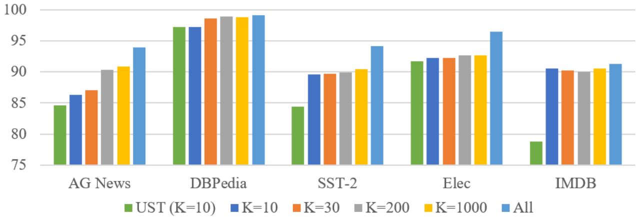

Figure 4: Test accuracy (%) under different number (K) of labeled training data per class.

Dataset	**BERT**	**UST**	**CEST(Ours)**
AG News	81.66	84.63	**86.22**
DBpedia	95.40	**97.21**	97.16
IMDB	75.40	78.83	**90.65**
Elec	76.69	91.76	**92.21**
SST-2	79.44	84.00	**89.56**
Average	81.72 (+0.00%)	87.29 (+5.57%)	**91.16** (+9.44%)

Table 4: Test accuracy (%) with only ten labels per class.

contributes to the performance, as the regularization explicitly enforces the smoothness assumption in SSL and betters the quality of features. As indicated in the table, constructing the reliable sub-graph increases around 0.58% accuracy (91.04% - 90.46%) because it mitigates the confusion caused by regularizing on the wrong edges.

Data Efficiency of CEST

To explore the limit of our framework, we examine our framework with only ten labeled training data per class, *i.e.,* 10, 30 labeled data per class for training and validation dataset, respectively. Under such label-limited settings, the results in Table 4 show that CEST still outperforms UST by a large margin, even larger than the case with 30 labels per class for training. The higher performance can be attributed to the use of reliable similarity graph. The graph enables CEST to exploit the underlying relationships among the few data, instead of using data separately, and to comply with smoothness assumption for better generalization capability.

In Fig. 4, we run our framework under different numbers (K) of labels per class, $K = \{10, 30, 200, 1000\}$. We observe that the accuracy gradually improves as K increases, as expected. It is noteworthy that the performance difference between using ten labeled training data per class and using more labeled training data is small, indicating that our framework has higher data efficiency indeed. Finally, in Table 5, we compare CEST with more non-BERT-based SSL approaches that use different labels. Our framework demonstrates large performance gain, especially on IMDB dataset with at least 8%, while using 3 $\sim$ 6 times fewer labeled data.

Datasets	Model	# of labels	Acc.
IMDB	CEST (ours)	30	90.2
	Variational Pre-training	200	82.2
	RL + Adv. Training	100	82.1
	SeqSSL + Self-training	100	79.7
	Layer Part. + Temp. Ens.	100	75.8
	SeqSSL + Adv. Training	100	75.7
	Layer Part. + Π model	100	69.3
AG News	CEST (ours)	30	87.1
	Variational Pre-training	200	83.9
	SeqSSL + Self-training	100	78.5
	SeqSSL + Adv. Training	100	73.0
DBpedia	CEST (ours)	30	98.6
	RL + Adv. Training	100	98.5
	SeqSSL + Self-training	100	98.1
	SeqSSL + Adv. Training	100	96.1

Table 5: Performance comparison with non-BERT-based semi-supervised approaches. (Li and Ye 2018; Gururangan et al. 2019; Dai and Le 2015; Li and Sethy 2020) (RL: Reinforcement Learning, Adv.: Adversarial, Temp. Ens.: Temporal Ensemble, Layer Part.: Layer Partitioning)

Conclusion

In this work, we introduce Contrast-Enhanced Semi-supervised Text classification, CEST, under label-limited settings. CEST judiciously selects unlabeled data for self-training and leverages reliable similarity graph to consider the smoothness in the feature space. It mitigates confirmation bias by proposing a new formulation and by using the noise-robust loss. Through experiments on five benchmark datasets, CEST shows strong performance over state-of-the-art semi-supervised learning approaches. Some interesting future works include extending this method to more severe low-resource settings or integrating it into real-world applications, such as healthcare applications.

Acknowledgements

This research was supported by the Joint Research Center for AI Technology and All Vista Healthcare under the Ministry of Science and Technology of Taiwan, under the grant numbers of 110-2634-F-002-042- , 110-2634-F-002-016- , 110-2634-F-002-046- and 110-2634-F-002-049-, as well as Center for Artificial Intelligence and Advanced Robotics, National Taiwan University.

References

Arazo, E.; Ortego, D.; Albert, P.; O'Connor, N. E.; and McGuinness, K. 2020. Pseudo-labeling and confirmation bias in deep semi-supervised learning. In *2020 International Joint Conference on Neural Networks (IJCNN)*, 1–8. IEEE.

Chawla, N. V.; and Karakoulas, G. 2005. Learning from labeled and unlabeled data: An empirical study across techniques and domains. *Journal of Artificial Intelligence Research*, 23: 331–366.

Chen, J.; Yang, Z.; and Yang, D. 2020. 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*, 2147–2157. Online: Association for Computational Linguistics.

Chen, M.; Tang, Q.; Livescu, K.; and Gimpel, K. 2018. Variational Sequential Labelers for Semi-Supervised Learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 215–226. Brussels, Belgium: Association for Computational Linguistics.

Dai, A. M.; and Le, Q. V. 2015. Semi-supervised Sequence Learning. In Cortes, C.; Lawrence, N. D.; Lee, D. D.; Sugiyama, M.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, 3079–3087.

Devlin, J.; Chang, M.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Burstein, J.; Doran, C.; and Solorio, T., eds., *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, 4171–4186. Association for Computational Linguistics.

Dror, R.; Baumer, G.; Shlomov, S.; and Reichart, R. 2018. The Hitchhiker's Guide to Testing Statistical Significance in Natural Language Processing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1383–1392. Melbourne, Australia: Association for Computational Linguistics.

Gal, Y.; and Ghahramani, Z. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, 1050–1059. PMLR.

Gal, Y.; Islam, R.; and Ghahramani, Z. 2017. Deep Bayesian Active Learning with Image Data. In Precup, D.; and Teh, Y. W., eds., *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, 1183–1192. PMLR.

Grandvalet, Y.; and Bengio, Y. 2004. Semi-Supervised Learning by Entropy Minimization. In *Proceedings of the 17th International Conference on Neural Information Processing Systems*, NIPS'04, 529–536. Cambridge, MA, USA: MIT Press.

Gururangan, S.; Dang, T.; Card, D.; and Smith, N. A. 2019. Variational Pretraining for Semi-supervised Text Classification. In Korhonen, A.; Traum, D. R.; and Màrquez, L., eds., *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, 5880–5894. Association for Computational Linguistics.

Houlsby, N.; Huszár, F.; Ghahramani, Z.; and Lengyel, M. 2011. Bayesian Active Learning for Classification and Preference Learning. *ArXiv*, abs/1112.5745.

Kim, Y. 2014. Convolutional Neural Networks for Sentence Classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1746–1751. Doha, Qatar: Association for Computational Linguistics.

Kumar, A.; Ma, T.; and Liang, P. 2020. Understanding Self-Training for Gradual Domain Adaptation. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, 5468–5479. PMLR.

Lee, D.-H.; 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.

Li, A. H.; and Sethy, A. 2020. Semi-Supervised Learning for Text Classification by Layer Partitioning. In *2020 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2020, Barcelona, Spain, May 4-8, 2020*, 6164–6168. IEEE.

Li, Y.; and Ye, J. 2018. Learning Adversarial Networks for Semi-Supervised Text Classification via Policy Gradient. In Guo, Y.; and Farooq, F., eds., *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018*, 1715–1723. ACM.

Luo, Y.; Zhu, J.; Li, M.; Ren, Y.; and Zhang, B. 2018. Smooth neighbors on teacher graphs for semi-supervised learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 8896–8905.

Maas, A. L.; Daly, R. E.; Pham, P. T.; Huang, D.; Ng, A. Y.; and Potts, C. 2011. Learning Word Vectors for Sentiment Analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 142–150. Portland, Oregon, USA: Association for Computational Linguistics.

Maaten, L. V. D.; and Hinton, G. E. 2008. Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9: 2579–2605.

McAuley, J.; and Leskovec, J. 2013. Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proceedings of the 7th ACM conference on Recommender systems*, 165–172.

Mendes, P.; Jakob, M.; and Bizer, C. 2012. DBpedia: A Multilingual Cross-domain Knowledge Base. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, 1813–1817. Istanbul, Turkey: European Language Resources Association (ELRA).

Meng, Y.; Zhang, Y.; Huang, J.; Xiong, C.; Ji, H.; Zhang, C.; and Han, J. 2020. Text Classification Using Label Names Only: A Language Model Self-Training Approach. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 9006–9017. Online: Association for Computational Linguistics.

Menon, A. K.; Rawat, A. S.; Reddi, S. J.; and Kumar, S. 2020. Can gradient clipping mitigate label noise? In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Miyato, T.; Dai, A. M.; and Goodfellow, I. J. 2017. Adversarial Training Methods for Semi-Supervised Text Classification. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Miyato, T.; Maeda, S.-i.; Koyama, M.; and Ishii, S. 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.

Mukherjee, S.; and Awadallah, A. H. 2020. Uncertainty-aware Self-training for Few-shot Text Classification. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Sennrich, R.; Haddow, B.; and Birch, A. 2016. Improving Neural Machine Translation Models with Monolingual Data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 86–96. Berlin, Germany: Association for Computational Linguistics.

Socher, R.; Perelygin, A.; Wu, J.; Chuang, J.; Manning, C. D.; Ng, A.; and Potts, C. 2013. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, 1631–1642. Seattle, Washington, USA: Association for Computational Linguistics.

Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1): 1929–1958.

Tang, D.; Qin, B.; and Liu, T. 2015. Document Modeling with Gated Recurrent Neural Network for Sentiment Classification. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 1422–1432. Lisbon, Portugal: Association for Computational Linguistics.

Tarvainen, A.; and Valpola, H. 2017. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *NIPS*.

Wang, H.; and Yeung, D.-Y. 2016. Towards Bayesian deep learning: A framework and some existing methods. *IEEE Transactions on Knowledge and Data Engineering*, 28(12): 3395–3408.

Xie, Q.; Dai, Z.; Hovy, E. H.; Luong, T.; and Le, Q. 2020. Unsupervised Data Augmentation for Consistency Training. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Yang, Z.; Hu, Z.; Salakhutdinov, R.; and Berg-Kirkpatrick, T. 2017. Improved variational autoencoders for text modeling using dilated convolutions. In *International conference on machine learning*, 3881–3890. PMLR.

Yang, Z.; Yang, D.; Dyer, C.; He, X.; Smola, A.; and Hovy, E. 2016. Hierarchical Attention Networks for Document Classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 1480–1489. San Diego, California: Association for Computational Linguistics.

Zhang, C.; Bengio, S.; Hardt, M.; Recht, B.; and Vinyals, O. 2017. Understanding deep learning requires rethinking generalization. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Zhang, X.; Zhao, J.; and LeCun, Y. 2015. Character-level Convolutional Networks for Text Classification. In Cortes, C.; Lawrence, N.; Lee, D.; Sugiyama, M.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Zhang, Z.; and Sabuncu, M. R. 2018. Generalized Cross Entropy Loss for Training Deep Neural Networks with Noisy Labels. In Bengio, S.; Wallach, H. M.; Larochelle, H.; Grauman, K.; Cesa-Bianchi, N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, 8792–8802.