Proceedings of the Eighteenth International AAAI Conference on Web and Social Media (ICWSM 2024)

# SLaNT: A Semi-Supervised Label Noise-Tolerant Framework for Text Sentiment Analysis

**Bin Cao, Kai Jiang, Jing Fan***

Department of Computer Science, Zhejiang University of Technology, Hangzhou, China
{bincao, jiangkai, fanjing}@zjut.edu.cn

## Abstract

The exponential growth of user-generated comment data on social media platforms has greatly promoted research on text sentiment analysis. However, the presence of conflicting sentiments within user comments, known as 'user comments with noisy labels', poses a significant challenge to the reliability of text sentiment analysis models. Many current approaches address this issue by either discarding noisy samples or assigning small weights to them during training, but these strategies can lead to sample wastage and reduced model robustness. In this paper, we present SLaNT, a novel semi-supervised label noise-tolerant framework specifically designed for text sentiment analysis. SLaNT employs a four-module pipeline that includes Noisy Data Identification, Data Augmentation, Noisy Data Relabeling, and Re-training. Notably, SLaNT introduces an early stopping strategy to efficiently identify noisy samples. Additionally, to mitigate confirmation bias during the relabeling of noisy data, a unique co-relabeling strategy based on ensemble learning is integrated into SLaNT. Experimental results on four text user comment datasets demonstrate that SLaNT significantly outperforms four selected strong baselines.

## Introduction

With the rapid advancement of social media technology, individuals can express their sentiments on various social platforms such as Twitter and Facebook. Text Sentiment analysis models play a pivotal role in discerning users' emotions by analyzing these comments. For instance, businesses can gauge the perceived quality of their products by examining customer feedback on the Yelp website. Unfortunately, as shown in Table 1, a substantial number of noisy labels exist in the extensive corpus of user comments. These noisy comments manifest as inconsistencies between the sentiments expressed in comments and the corresponding ratings, such as a positive comment (*It's one of the most beautiful hotels I've seen.*) but attached with a negative rating (*1*). This paradoxical phenomenon, aptly termed 'user comments with noisy labels', significantly impacts the accuracy and reliability of text sentiment analysis models (Zhang et al. 2021a).

Many Noise-tolerant learning (Kearns 1998; Tu et al. 2023; Song et al. 2022) methods have been proposed to ensure the robustness of model when training with noisy labels. These methods can be divided into three groups based on how the sample is processed: (1) **Data reweighting**, the data that has an unreliable label will be assigned with a small weight to reduce its contribution to minimization of the training loss (Sun et al. 2022a; Zhang et al. 2021b; Shu et al. 2019; Ren et al. 2018; Jiang et al. 2018). (2) **Data filtering**, noisy data will be identified and discarded before training occurs(Xia et al. 2021; Northcutt, Jiang, and Chuang 2021; Pleiss et al. 2020). (3) **Data relabeling**, where the noisy data is relabeled, and then together with clean data, both will be leveraged by the deep neural network (DNN) (Mallem, Hasnat, and Nakib 2023; Zheng, Awadallah, and Dumais 2021; Tu et al. 2023). Compared with the first two groups of methods, where only clean data plays an important role in modeling, methods in the third group can leverage the noisy samples to further improve the generalization performance. However, most data relabeling methods either assume the existence of a small set of clean data which is sometimes difficult to obtain in real-world scenarios (Anomaly 2022; Yan et al. 2016; Zheng, Awadallah, and Dumais 2021; Gong et al. 2022), or there exist modules closely bound to the image processing field (Zhou, Wang, and Bilmes 2020; Li, Socher, and Hoi 2020) and cannot be used for text sentiment analysis.

| Comments | Ratings | Noisy or Clean |
|---|---|---|
| The service of hotel is terrible! | 4 | noisy |
| I like this movie. | 3 | clean |

Table 1: Example of clean and noisy user comments. Ratings range from 0 to 4, with 0 indicating extremely negative and 4 indicating extremely positive.

In fact, given a corpus of noisy user comments, to design a label noise-tolerant text sentiment analysis model in a data relabeling manner, two sub-tasks should be carefully considered, i.e., *how to correctly identify the noisy data from the whole noisy dataset?* and *how to relabel the noisy data with the correct ones?* Though there already exist some solutions for these two sub-tasks respectively, their connection hasn't been well explored so far for the task of sentiment analysis.

In this paper, we propose SLaNT, a Semi-supervised Label Noise-Tolerant learning framework for text sentiment anal-

*Corresponding author.

ysis. First, the mislabeled data is identified through Confident Learning (CL) (Northcutt, Jiang, and Chuang 2021) which can effectively measure the label quality for the noisy dataset by combining different principles of noisy data processing. However, due to the memorization effect (Zhang et al. 2021a), DNN model will overfit noisy data, subsequently impacting the performance of CL. To mitigate this issue, we introduce an early stopping strategy. Compared with other techniques for identifying noise, e.g., the Gaussian Mixture Model (GMM) (Li, Socher, and Hoi 2020) and confidence value strategy (Bai et al. 2021), CL with our early stopping strategy needs no hyper-parameter for identifying noisy data and has better performance. Then, a semi-supervised learning (SSL) strategy is used to correct the noisy data and train the model. The reason is that the label-noise tolerant problem can be transformed to SSL if we treat noisy data as *unlabeled* data and the rest as *labeled* data.

Based on the above two main techniques, SLaNT can help to get a robust DNN model by using the following four modules: (1) Noisy data identification module, where noisy training data is divided into clean data and noisy data through pre-training DNN models with early stopping and Confident Learning. (2) Data augmentation module, which uses various textual data augmentation techniques to expand the identified noisy data for the purpose of consistency regularization of SSL (Berthelot et al. 2019). (3) Noisy data relabeling module, where SLaNT first uses an ensemble learning strategy to predict the pseudo-label for the noisy data then co-relabel them based on the combination of its original label and the pseudo label. (4) Re-training module, where SLaNT mixes both clean data and re-labeled noisy data to retrain the model. Our contributions can be summarized as follows:

- To counteract the noisy labels in user comments for sentiment analysis, we propose a semi-supervised learning framework SLaNT which can help to build a noise-tolerant DNN model by leveraging both clean and noisy data.
- To prevent the DNN model from overfitting to noise and facilitate efficient identification of noisy samples in the first module, we propose using an early stopping strategy to train DNN models before applying CL.
- To relabel the noisy data reliably, we not only use ensemble learning to predict the pseudo labels to avoid the confirmation bias of a single model but also co-relabel the noisy data by combining the original label with the pseudo label.
- To validate the performance of SLaNT, we conduct extensive experiments on four noisy user comment datasets, and the results show that SLaNT can obtain significant improvement in accuracy when compared with four strong baselines.

The remainder of this paper is organized as follows. Section  presents the detailed implementation of SLaNT. Section  reports the results of experimental evaluation. The related work is reviewed in Section . Section  concludes the paper.

# SLaNT Implementation

Figure 1 shows our SLaNT framework where four modules are included, i.e., Noisy Data Identification, Data Augmentation, Noisy Data Relabeling, and Re-training. It is important to note that although $L$ DNN models are involved in SLaNT when performing steps of pre-training, early stopping and ensemble learning, a great amount of training time can be saved by executing these steps in a parallel manner.

## Noisy Data Identification

The purpose of this module is to find noisy labeled samples as many as possible. To this end, SLaNT firstly pre-trains the DNN model (e.g., BERT(Devlin et al. 2019)) so that the model could learn the distribution of training data(Gururangan et al. 2020).

**Early stopping.** Note that to avoid overfitting the noisy labels in the late stage of training (Arpit et al. 2017), SLaNT adopts an early stopping strategy to terminate the optimization at the early stage of training for the DNN model (Nguyen et al. 2020; Bai et al. 2021). Inspired by the work of Bai et al. (Bai et al. 2021), we understand that noise has a more adverse effect on the latter layers than the former layers of the network. So we introduce an early stopping strategy where we first train the whole network with a relatively small number of epochs, and then only optimize the former layers by fixing the last output layer to mitigate the influence of noise.

After optimizing with early stopping, the trained DNN model will be further used for generating a probability matrix by performing prediction for each sample. Next, based on the probability matrix, SLaNT adopts the component of Confident Learning (CL) to divide training data $X$ into clean samples and noisy samples. In fact, according to the theoretical proof in CL (Northcutt, Jiang, and Chuang 2021), CL is able to find label errors exactly even when samples' predicted probabilities are erroneous, which again ensures the safety of adopting an early stopping strategy.

CL assumes that each sample in the training data $X$ has a latent true label $y^*$, and due to the so-called class-conditional classification noise process, the true label $y^*$ may be independently mislabeled as the observed label $\tilde{y}$ with a certain probability. This assumption is commonly used in other works (Sukhbaatar et al. 2015; Goldberger and Ben-Reuven 2017). Based on the assumption, two inputs are required in CL: (1) the $m \times n$ matrix $\hat{P}$, where $n$ and $m$ denote the number of training samples and labels respectively, and each element $\hat{P_{k,i}}$ indicates the predicted probability of $i$th training sample belonging to $k$th label. (2) $\tilde{y}$, one-hot encoding over $m$ original labels. Note that, the probabilities within $\hat{P}$ can be computed by the pre-trained DNN models like BERT(Devlin et al. 2019), Roberta(Liu et al. 2019), etc.

CL estimates the matrix $\hat{P}$ to identify noisy data. The pre-trained model $\theta$ affects $\hat{P}$ which in turn affects the performance of CL. The closer the probabilities within $\hat{P}$ are to the true probabilities of the training data, the better the performance of confident learning. Hence, SLaNT pre-trains the model $\theta$ to improve predicted probabilities that impart the knowledge of the training data distribution to the model $\theta$ without using the noisy labels. Subsequent experiments also

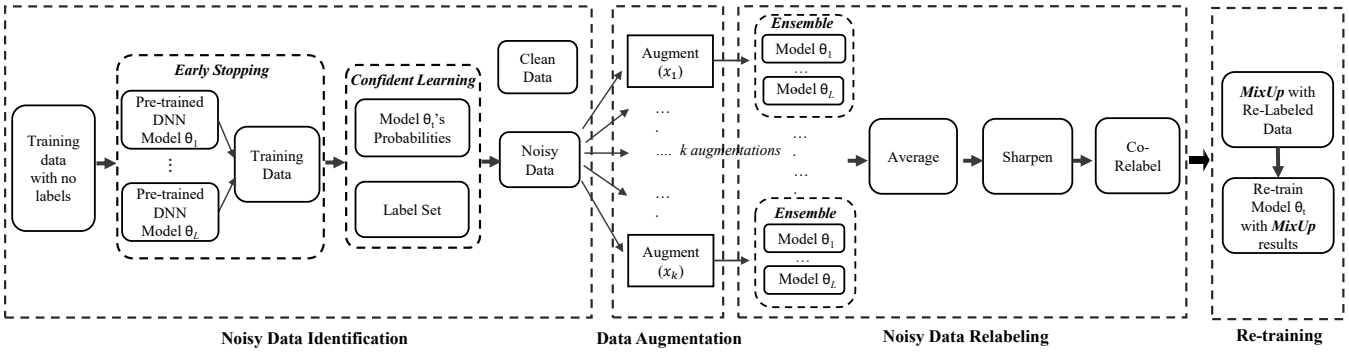



Figure 1: Overview of SLaNT framework.

prove that further pre-training DNN model can improve the performance of Confident Learning.

The main procedure of CL comprises the following two steps:

**Step 1. Count: Characterize and Find Label Errors using the Confident Joint.** CL uses confident joint $C_{\tilde{y},y^*}$ to partition and count noisy labels. The definition of the confident joint is as follows:

$$C_{\tilde{y},y^*}[i][j] := \left|\hat{X}_{\tilde{y}=i,y^*=j}\right| \quad \text{where}$$
$$\hat{X}_{\tilde{y}=i,y^*=j} := \{x \in X_{\tilde{y}=i} : \hat{p}(\tilde{y}=j;x,\theta) \geq t_j\} \tag{1}$$

$\hat{X}_{\tilde{y}=i,y^*=j}$ is the set of samples $x$ labeled $\tilde{y}=i$ with large enough $\hat{p}(\tilde{y}=j;x,\theta)$ to likely belong to class $y^*=j$, determined by a per-class threshold $t_j$. $y^*$ represents the true label, and the threshold $t_j$ is the average of model's predicted probabilities for each class:

$$t_j = \frac{1}{|X_{\tilde{y}=j}|} \sum_{x \in X_{\tilde{y}=j}} \hat{p}(\tilde{y}=j;x,\theta) \tag{2}$$

**Step 2. Rank and Prune: Data Cleaning.** Following the estimation of $C_{\tilde{y},y^*}$, SLaNT directly uses the sets of samples counted in the off-diagonals of $C_{\tilde{y},y^*}$ to estimate noisy labels as $\{x \in \hat{X}_{\tilde{y}=i,y^*=j} : i \neq j\}$.

Through the above two steps, SLaNT divides the noisy training data ($X$) into a clean labeled dataset ($C$) and a noisy labeled dataset ($U$) with label-confidence $w$. For each sample, label-confidence represents the probability $w_i(\tilde{y}=j;x_i,\theta)$ of sample $x_i$ belonging to its original label when considering the model parameters ($\theta$). Low label-confidence is a heuristic likelihood of being a label error.

## Data Augmentation for Textual Content

To correct the noisy samples, SLaNT adopts the idea of semi-supervised learning (SSL) by viewing the noisy data as unlabeled data. In much recent work for SSL, adding a loss term on unlabeled data can help the model to generalize better to unseen data, and one of the popular strategies for this loss term is consistency regularization(Berthelot et al. 2019) which commonly applies data augmentation to achieve the regularization purpose. Specifically, the label of the data after data augmentation is the same as the original label. In other words, sample $x$ should be classified the same as its augmentation $Augment(x)$.

Existing methods of textual data augmentation can be divided into three groups: paraphrasing, noising and sampling (Li, Hou, and Che 2021; Feng et al. 2021). The paraphrasing-based methods make some changes to the words, phrases and sentence structure which retain the texts' original semantics (Zhang, Zhao, and LeCun 2015; Hou et al. 2018). The noising-based methods add some discrete or continuous noise to texts which has little effect on semantics (Wei and Zou 2019; Coulombe 2018). The sampling-based methods sample novel data under the current data distributions (Kang et al. 2018; Du et al. 2021).

However, unlike image data augmentation, where elastically deforming or adding noise to an input image won't alter its label (Cireşan et al. 2010). Improper augmentation of text can easily alter the original label, which will add extra noise. For instance, if we randomly drop words from "*I want to inquire about mobile phone bills.*" to "*I want to inquire about mobile phone.*", the original meaning has been obviously changed. Therefore, we merely use the paraphrasing-based and the noising-based methods in SLaNT. For each $u_i$ in the batch of noisy labeled data $U$, we generate $K$ augmentations $\hat{u}_{i,k} = Augment(u_i), k \in (1,...,K)$ (Algorithm 1, line 12) through above textual data augmentation methods.

## Noisy Data Relabeling

Having acquired noisy labeled data $U$ and their augmented data $\bar{U}$, SLaNT uses the similar idea of MixMatch (Berthelot et al. 2019) to relabel the noisy samples $U$. MixMatch computes the average of the predicted class distributions $\bar{q}_i$ across all the $K$ augmentations of $\bar{u}_i$ as the new label for $u_i$ by:

$$\bar{q}_i = \frac{1}{K} \sum_{k=1}^{K} \mathrm{p}_{\mathrm{model}}\left(y \mid \bar{u}_{i,k};\theta\right) \tag{3}$$

However, $\bar{q}_i$ is derived from a single model, which has a confirmation bias problem that results in a deviation between predicted probability and true probability of the sample (Tarvainen and Valpola 2017). SLaNT makes two improvements

to MixMatch.

First, to reduce the confirmation bias of a single model, we perform ensemble learning with the $L$ weak classifiers obtained after early stopping in the module of noisy data identification. Specifically, SLaNT averages the $L$ models' predicted class distributions across all the $K$ augmentations of $\bar{u}_i$ by:

$$\bar{q}_i = \frac{1}{K * L} \sum_{k=1}^{K} \sum_{l=1}^{L} \mathrm{p}_{\mathrm{model_l}} \left( y \mid \bar{u}_{i,k}; \theta \right) \tag{4}$$

Second, we co-relabel for noisy labeled sample by linearly combining the noisy label $\tilde{y_i}$ with the $\bar{q}_i$ guided by the label-confidence $w_i$. The new label of noisy sample $\bar{y}_i$ is as follows:

$$\bar{y}_i = w_i \tilde{y_i} + (1 - w_i) \bar{q}_i \tag{5}$$

$\bar{y}_i$ is an average prediction over $L$ models for the $i^{th}$ sample. To reduce the entropy of the label distribution, we also apply a sharpening function(Zhang et al. 2021b):

$$\hat{y}_i = \mathrm{Sharpen}\left(\bar{y}_i, T\right) = \bar{y}_i^T \Big/ \sum_{j=1}^{m} \bar{y}_j^T, \text{ for } j = 1, 2, \ldots, m \tag{6}$$

where $T$ is a hyperparameter, $\bar{y}_i$ is the guessed label. When $T \to 0$, $Sharpen(\bar{y}_i, T)$ will approach a 'one-hot' distribution. $m$ is the number of labels.

## Re-training with Relabeled Data

To alleviate the potential errors propagated from the modules of noisy data identification and noisy data relabeling, SLaNT further uses the idea of MixUp (Zhang et al. 2018) to retrain the DNN model on both clean data and noisy data with the guessed pseudo label. The main idea of MixUp is as follows, for a pair of samples$(x_1, y_1)$,$(x_2, y_2)$, $(x', y')$ is computed by:

$$\lambda \sim \mathrm{Beta}(\alpha, \alpha) \tag{7}$$

$$\lambda' = \max(\lambda, 1 - \lambda) \tag{8}$$

$$x' = \lambda' x_1 + (1 - \lambda') x_2 \tag{9}$$

$$y' = \lambda' y_1 + (1 - \lambda') y_2 \tag{10}$$

where $\alpha$ is a hyperparameter. SLaNT combines clean labeled data $C$ with relabeled noisy data $\hat{U}$ and shuffles to form $W$, which can be viewed as the corrected version of training data $X$:

$$C = \left( (c_i, p_i) ; i \in (1, \ldots, |C|) \right) \tag{11}$$

$$\hat{U} = \Big( (\hat{u}_{i,k}, q_i) ; i \in (1, \ldots, |\hat{U}|), k \in (1, \ldots, K) \Big) \tag{12}$$

$$W = Shuffle(Concat(C, \hat{U})) \tag{13}$$

For each the $i^{th}$ sample-label pair in $C$, SLaNT computes $MixUp(C_i, W_i)$ and adds the result to $C'$. Similarly, based on the remainder of $W$, $MixUp(\hat{U}_i, W_{i+|C|})$ is computed and its result is added to $U'$.

| Dataset | Class | Train | Test | Dev |
|---|---|---|---|---|
| SST-2 | 2 | 7k | 2k | 1k |
| SST-5 | 3 | 7k | 2k | 1k |
| Amazon-5 | 5 | 2100k | 600k | 300k |
| Yelp-5 | 5 | 490k | 140k | 70k |

Table 2: The statistics of datasets used.

**Loss Function.** Unlike the cross-entropy loss, L2 loss is bounded and less sensitive to incorrect predictions(Brier et al. 1950). So, SLaNT uses L2 loss on $U'$ and cross-entropy loss on $C'$:

$$\mathcal{L}_{\mathcal{C}} = \frac{1}{|\mathcal{C}'|} \sum_{c,p \in \mathcal{C}'} \mathrm{H}\left(p, \mathrm{p}_{\mathrm{model}}\left(y \mid c; \theta\right)\right) \tag{14}$$

$$\mathcal{L}_{\mathcal{U}} = \frac{1}{|U'|} \sum_{u,q \in \hat{U}'} \left\| q - \mathrm{p}_{\mathrm{model}}\left(y \mid u; \theta\right) \right\|_2^2 \tag{15}$$

The total loss is:

$$\mathcal{L} = \mathcal{L}_{\mathcal{C}} + \lambda_{\mathcal{U}} \mathcal{L}_{\mathcal{U}} \tag{16}$$

where $\lambda_{\mathcal{U}}$ is the unsupervised loss weight.

# Experiments

In this section, we describe in detail the extensive experiments performed to evaluate the effectiveness of the proposed framework for text sentiment analysis with noisy labels.

## Datasets

We use four text user comment datasets to conduct sentiment analysis tasks, as listed in Table 2:

- **SST**: The Stanford Sentiment Treebank (SST) is a widely used collection of datasets that consists of single sentence movie reviews (Socher et al. 2013). Two of the most prominent datasets within SST are **SST-2** and **SST-5**. We use SST-2 for binary classification and SST-5 for multiple classification.
- **Amazon-5**: Amazon-5 is a popular dataset of user reviews collected from the Amazon website (McAuley and Leskovec 2013).
- **Yelp-5**: The Yelp-5 dataset is a local directory service with user reviews and it detects fine-grained sentiment labels (Zhang, Zhao, and LeCun 2015).

## Noise Settings

Following previous works (Han et al. 2018; Tanaka et al. 2018; Reed et al. 2015), we experiment with two types of label noise: **symmetric noise**, **asymmetric noise**. As shown in Figure 2, when the noise ratio is 0.5, the original label in the symmetric noise can flip to any class with the probability of 0.1. While in asymmetric noise, the original label can only flip to one of the rest classes with a probability of 0.5.

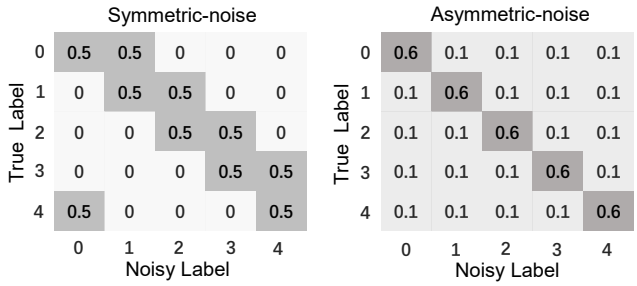



Figure 2: The noise transition matrix of symmetric and asymmetric noise with noise ratio 50%.

Previous studies (Han et al. 2018) have demonstrated that it is hard to learn a good model for asymmetric noise larger than 50%. Therefore, we evaluate our framework with noise rates 20% - 80% (for symmetric noise) and 10% - 40% (for asymmetric noise). Note that SST-2 is a binary classification dataset, we experiment with it only for symmetric noise from 0.1 to 0.4 with step 0.1.

## Baseline Methods

We compare our proposed framework SLaNT with the following four strong baselines:

- **DivideMix**, which dynamically fits a Gaussian Mixture Model (GMM) to identify noisy data and leverages SSL to correct noisy labels (Li, Socher, and Hoi 2020).
- **Confident Learning (CL)**, which identifies and removes noisy data before training the model (Northcutt, Jiang, and Chuang 2021).
- **MW-Net**, which reweights noisy data through a meta reweight net (Shu et al. 2019).
- **Co-meta**, which utilizes a meta-learning framework for data correction (Lee, Kim, and gyun Seo 2022).

## Experiment Settings and Parameter Tuning

SLaNT is implemented by PyTorch (version 1.6.0) (Paszke et al. 2019) and evaluated on a GeForce RTX 3090 Ti GPU. All results are averaged over 5 independent runs (Wu et al. 2021; Zhou, Wang, and Bilmes 2020).

**Models**. Three DNN models ($L = 3$) are ensembled in SLaNT: a 12-hidden_layer, 768-hidden_size Bert-Base-Uncased model, a 12-hidden_layer, 768-hidden_size Roberta model, and a 12-layer, 64-head XLNet (Yang et al. 2019). Among these, the Bert-Base-Uncased model is for predicting the probability matrix required by CL and re-training. The optimization function is AdamW with a momentum of 0.9, and a weight decay of 0.0005.

**Data augmentation**. We use three textual data augmentations ($K = 3$):

- **Dropout**, which generates different sentence vector representations by changing the dropout parameters of the fully connected layer (Srivastava et al. 2014).
- **CBert**, which generates a wider range of substitute words by using words predicted by a bi-directional language model (LM) according to the context (Wu et al. 2019).

| Dataset | Symmetric | | | | Asymmetric | |
|---|---|---|---|---|---|---|
| | 0.1 | 0.2 | 0.3 | 0.4 | 0.2 | 0.4 |
| SST-2 | 0 | 25 | 25 | 150 | - | - |
| | Symmetric | | | | Asymmetric | |
| | 0.2 | 0.4 | 0.6 | 0.8 | 0.2 | 0.4 |
| SST-5 | 50 | 150 | 150 | 150 | 25 | 150 |
| Amazon-5 | 50 | 50 | 150 | 150 | 25 | 150 |
| Yelp-5 | 25 | 50 | 150 | 150 | 25 | 50 |

Table 3: $\lambda_{\mathcal{U}}$ used on four datasets.

- **Text Gen**, a novel text generation approach for long and short text (Bayer et al. 2023).

**Tuning for $\lambda_{\mathcal{U}}$**. Following the work of MixUp (Zhang et al. 2018), we tune the $\lambda_u \in \{0,\ 25,\ 50,\ 150\}$ by fixing $T = 0.5$ and $\alpha = 0.5$. For all datasets with the noise rate varying from low to high, we find that a higher noise ratio requires a larger $\lambda_{\mathcal{U}}$. Hence, we finally use different $\lambda_u$ for four datasets listed in Table 3.

**Other hyperparameters: $T$ and $\alpha$**. By tuning $T$ in the range $[0.2, 0.6]$ and $\alpha$ in the range $[0, 1]$ with step 0.1, we find SLaNT can achieve relatively good results for all experiments when setting $T = 0.5$ and $\alpha = 0.5$.

## Ablation Study

Here, we provide an analysis of our proposed framework SLaNT by removing different components on four datasets. We analyze the results in Table 4 as follows.

Overall, the fully equipped SLaNT outperforms other variants for all test cases, which demonstrates the effectiveness of integrating all the proposed components. Specifically, for different SLaNT variants:

- ***w/o augmentation***, where we use no augmentation to predict pseudo-label, the decrease in accuracy suggests that using data augmentation is beneficial for SLaNT. Moreover, we can observe that up to 2.32% can be obtained on SST-5 dataset with symmetric-0.8 noise, respectively.
- ***w/o ensemble***, where we use a single model to predict the pseudo label for identified noisy samples. The results suggest that the ensemble of multiple models is better than merely using a single one. The most significant improvements are gained on Yelp-5 dataset with up to 2.63% increase in accuracy for symmetric-0.8 noise..
- ***w/o label-confidence***, where we relabel the noisy sample without considering the original label, i.e., set the label-confidence $w$ to 0 in Equation 5. We find that considering label-confidence for noisy sample relabeling is effective for SLaNT, and the best improvements occur on Amazon-5 dataset, i.e., SLaNT increases the accuracy by up to 2.19%.

| Dataset | Method/Noise | Symmetric | | | | Asymmetric | |
|---|---|---|---|---|---|---|---|
| | | 0.1 | 0.2 | 0.3 | 0.4 | 0.2 | 0.4 |
| SST-2 | SLaNT w/o augmentation | 84.31 | 82.12 | 80.41 | 51.84 | - | - |
| | SLaNT w/o ensemble | 83.87 | 81.71 | 79.09 | 51.16 | - | - |
| | SLaNT w/o label-confidence | 84.72 | 82.29 | 79.92 | 51.25 | - | - |
| | SLaNT w/o Mix Up | 84.64 | 82.51 | 80.78 | 51.53 | - | - |
| | SLaNT | **85.28** | **83.64** | **82.06** | **52.47** | - | - |
| | | Symmetric | | | | Asymmetric | |
| | | 0.2 | 0.4 | 0.6 | 0.8 | 0.2 | 0.4 |
| SST-5 | SLaNT w/o augmentation | 50.07 | 47.12 | 38.25 | 25.26 | 50.16 | 48.37 |
| | SLaNT w/o ensemble | 48.93 | 45.81 | 37.24 | 25.70 | 49.04 | 45.43 |
| | SLaNT w/o label-confidence | 49.57 | 47.63 | 37.72 | 24.60 | 49.81 | 48.35 |
| | SLaNT w/o Mix Up | 50.11 | 46.95 | 37.28 | 26.04 | 49.53 | 48.10 |
| | SLaNT | **51.32** | **48.96** | **40.25** | **27.58** | **51.73** | **50.14** |
| Yelp-5 | SLaNT w/o augmentation | 60.39 | 58.13 | 51.06 | 36.54 | 60.52 | 58.51 |
| | SLaNT w/o ensemble | 60.45 | 58.07 | 50.63 | 36.32 | 60.39 | 57.94 |
| | SLaNT w/o label-confidence | 61.53 | 59.26 | 50.97 | 37.20 | 61.74 | 59.23 |
| | SLaNT w/o Mix Up | 60.17 | 58.87 | 52.34 | 36.11 | 60.23 | 58.92 |
| | SLaNT | **61.68** | **59.72** | **53.26** | **38.19** | **61.97** | **60.04** |
| Amazon-5 | SLaNT w/o augmentation | 59.58 | 55.98 | 49.23 | 38.57 | 59.92 | 56.27 |
| | SLaNT w/o ensemble | 59.72 | 56.19 | 50.77 | 39.04 | 60.38 | 55.32 |
| | SLaNT w/o label-confidence | 60.63 | 56.34 | 50.04 | 39.43 | 61.07 | 56.84 |
| | SLaNT w/o Mix Up | 60.14 | 56.17 | 50.83 | 38.25 | 60.43 | 56.56 |
| | SLaNT | **60.95** | **57.28** | **52.13** | **40.89** | **61.26** | **57.95** |

Table 4: Accuracy of ablation study on four datasets.

- ***w/o MixUp***, where we re-train relabeled noisy data without using MixUp. The degraded performance proves that combining clean and noisy data for re-training is effective. On SST-2 dataset, SLaNT has the most obvious improvement, i.e., up to 1.28% for accuracy.

## Comparison with Baseline Methods

We compare SLaNT with baselines on four user comment datasets and report their results in Table 5. Generally, SLaNT achieves the best performance for all test cases and noise rates. For example, even on the high noise rates (Yelp-5 dataset with symmetric-0.6 noise), our method also achieves better classification results (53.26%). Specifically, SLaNT outperforms MW-Net by up to 5.27% and 4.36% of accuracy on SST-5 and Amazon-5 datasets with symmetric-0.8 noise, respectively. This is because MW-Net has no mechanism for correcting noisy labels where it only assigns small weights to noisy samples. Similarly, the main reason for CL's poor performance is that it discards too many noisy samples when the noise ratio is high which reduces the performance of DNNs. A discernible trend observed from the outcomes on both Yelp-5 and Amazon-5 datasets suggests that the efficacy of Co-meta experiences a decline with an increase in the noise ratio. This phenomenon can be attributed to its estimated parameter update method, which inadvertently leads the label corrector to conform to the noise in the data, thereby compromising overall performance.

Moreover, compared with the results on SST5 dataset, the advantage of SLaNT is more obvious on Amazon-5 dataset. For example, SLaNT achieves 57.28% accuracy under 40% symmetric noise, which is significantly higher than that obtained by DivideMix (56.05%) and Co-meta (55.73%).

## Inside Study

Based on the accuracy metric, we now perform an inside study to further investigate how SLaNT achieves its effectiveness and mainly try to answer the following two questions: (1) *How the SSL strategy performs in SLaNT?* (2) *How can SLaNT perform better than DivideMix which also follows the similar SSL based pipeline?*

| | | Symmetric | | | | Asymmetric | |
|---|---|---|---|---|---|---|---|
| | | 0.1 | 0.2 | 0.3 | 0.4 | 0.2 | 0.4 |
| SST-2 | DivideMix | 84.86 | 83.32 | 81.12 | 51.24 | - | - |
| | Confident Learning | 83.54 | 81.95 | 77.64 | 46.82 | - | - |
| | MW-net | 83.97 | 82.12 | 81.05 | 50.39 | - | - |
| | Co-meta | 84.23 | 82.48 | 80.83 | 51.18 | - | - |
| | SLaNT | **85.28** | **83.64** | **82.06** | **52.47** | - | - |
| | | Symmetric | | | | Asymmetric | |
| | | 0.2 | 0.4 | 0.6 | 0.8 | 0.2 | 0.4 |
| SST-5 | DivideMix | 50.93 | 48.09 | 38.94 | 25.12 | 51.26 | 49.35 |
| | Confident Learning | 50.28 | 46.85 | 38.07 | 21.44 | 50.49 | 46.93 |
| | MW-net | 50.37 | 47.24 | 38.61 | 22.31 | 50.57 | 47.30 |
| | Co-meta | 50.51 | 47.53 | 38.59 | 22.46 | 50.82 | 48.52 |
| | SLaNT | **51.32** | **48.96** | **40.25** | **27.58** | **51.73** | **50.14** |
| Yelp-5 | DivideMix | 61.32 | 58.67 | 51.79 | 35.41 | 61.68 | 58.89 |
| | Confident Learning | 60.74 | 55.42 | 48.67 | 31.32 | 60.83 | 56.05 |
| | MW-net | 60.35 | 56.96 | 50.04 | 32.24 | 61.04 | 57.37 |
| | Co-meta | 61.29 | 58.34 | 51.23 | 34.05 | 61.35 | 58.51 |
| | SLaNT | **61.68** | **59.73** | **53.26** | **38.19** | **61.97** | **60.04** |
| Amazon-5 | DivideMix | 60.23 | 56.05 | 50.34 | 38.17 | 60.45 | 56.97 |
| | Confident Learning | 58.72 | 54.16 | 47.65 | 34.31 | 59.89 | 55.01 |
| | MW-net | 59.16 | 55.02 | 48.17 | 36.53 | 60.37 | 56.13 |
| | Co-meta | 60.34 | 55.73 | 49.58 | 38.20 | 60.92 | 56.86 |
| | SLaNT | **60.95** | **57.28** | **52.13** | **40.89** | **61.26** | **57.95** |

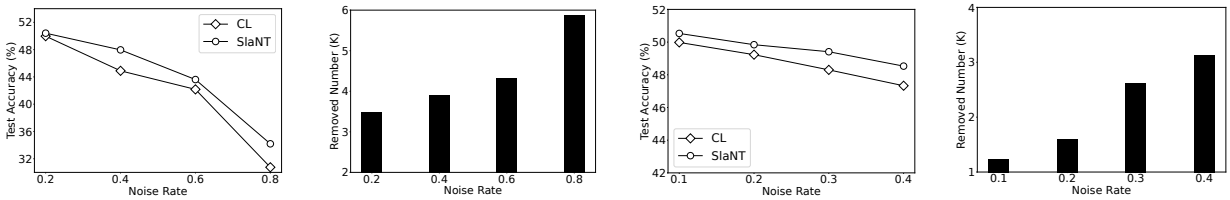
Table 5: Comparison with state-of-the-art baseline methods on four datasets.

For the first question, based on Table 5 we plot the accuracy of SLaNT and CL baseline in Figures 3(a), 3(c), 4(a), and 4(c). Note that, no label correction strategy is used in CL, i.e., the identified noisy samples are removed and the DNN model is trained only on the rest samples. So far, we can easily observe SLaNT performs better than CL, and this is because SLaNT which adopts the SSL strategy can fully utilize the noisy samples to achieve better generalization performance while CL cannot. To further study how SSL performs within SLaNT, we then plot the corresponding numbers of removed noisy samples in Figures 3(b), 3(d), 4(b), and 4(d). As shown in these figures, the number of removed noisy samples goes up as the noise ratio becomes high. For example, SLaNT uses more than 150K with symmetric-0.6 and 250K asymmetric-0.8 noisy samples than CL does on Yelp-5 dataset, while 6K around symmetric and 3k around asymmetric noisy samples are ignored by CL on SST-5 dataset.

Similarly, these above figures also suggest that MW-Net has underutilized the potential contribution of noisy samples to the model, as it assigns lower weights to the noisy samples, resulting in it being less effective than SLaNT. Consequently, it is reasonable to conclude that label correction methods generally outperform sample filtering and sample reweighting methods, especially in high-noise scenarios.

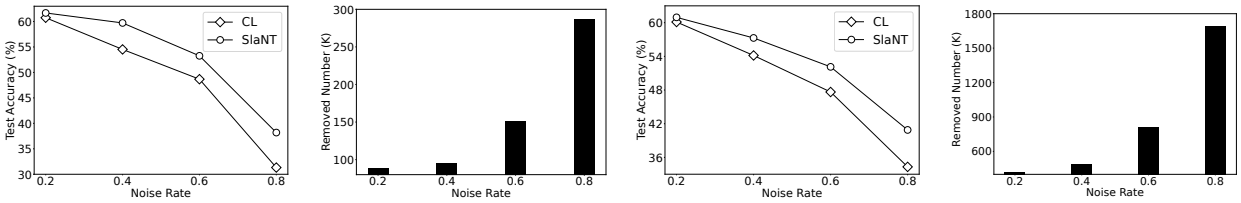
To answer the second question, we should be aware of two main differences between SLaNT and DivideMix: (1) The main component for identifying noisy data, i.e., DivideMix uses GMM while SLaNT adopts CL; (2) The relabeling strategy for noisy data, i.e., DivideMix uses two networks to perform label co-guessing without considering the information of original noisy label, while SLaNT relabels the noisy data by combining the original label and the pseudo label predicted by ensemble learning. Then we investigate as follows.

First, we compare the accuracy of SLaNT and DivideMix for identifying noisy data. As shown in Figure 5, SLaNT has higher accuracy than that of DivideMix, e.g. the improvement of accuracy is up to 5.27% on Amazon-5 with symmetric-0.8, which indicates that CL has better performance than that of GMM for identifying noisy data and less errors will be propagated to the following components in SLaNT.




(a) Accuracy-symmetric (b) Noisy samples removed-symmetric (c) Accuracy-asymmetric (d) Noisy samples removed-asymmetric

Figure 3: Inside study for SSL on SST-5. (a) and (c): accuracy of SLaNT and CL; (b) and (d): noisy samples removed for CL.




(a) Accuracy (b) Noisy samples removed (c) Accuracy (d) Noisy samples removed

Figure 4: Inside study for SSL on Yelp-5 and Amazon-5 with symmetric noise. (a) and (c): accuracy of SLaNT and CL on Yelp-5 and Amazon-5, respectively; (b) and (d): noisy samples removed for CL Yelp-5 and Amazon-5, respectively.

Second, we construct a new pipeline by combining CL with the relabel strategy (co-guessing) of DivideMix, denoted as *CL+DivideMix_relabel*, and report the comparison results in Figure 6. Obviously, SLaNT significantly outperforms the constructed baseline in all cases, which demonstrates that our relabeling strategy is more effective than that of DivideMix.

## Related Work

### Text Sentiment Analysis

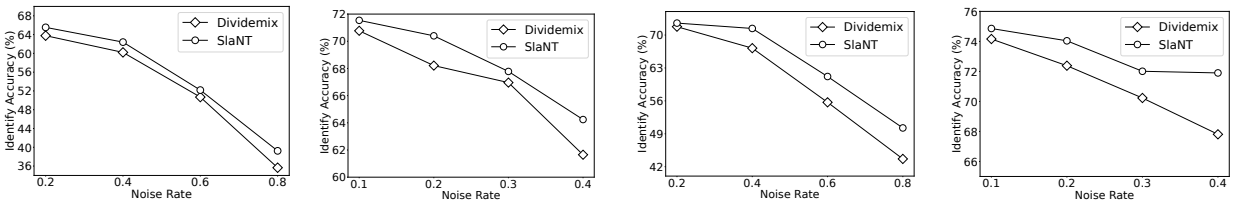
With the rise of social media platforms such as Twitter and Facebook, individuals now have the means to share their opinions with a global audience. Social media has emerged as one of the most significant modes of communication today. Consequently, a vast amount of associated text data has been generated, leading to the development of sentiment analysis technology for its analysis. Text sentiment analysis can be categorized into three levels: sentence level, document level, and aspect level, depending on the dataset. In this paper, we focus on conducting sentiment analysis based on sentence text. The objective of text sentiment analysis is to employ natural language processing techniques to assess the attitudes expressed in user-generated content, which can be positive, negative, or neutral.

### Noisy Label Learning

We note several prior works that have been developed against label noise.

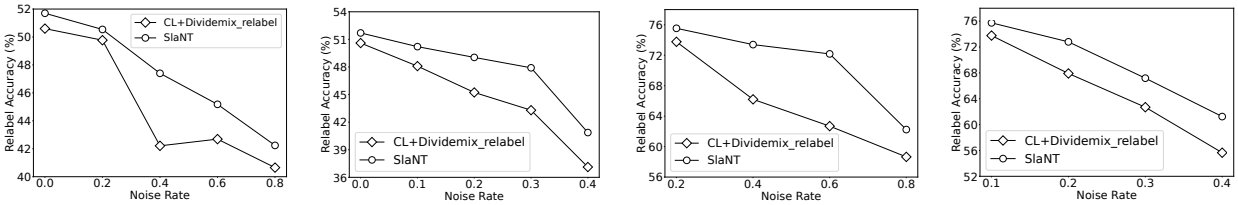
**Data reweighting.** Much existing work focused on reweighting the noisy data (Ren et al. 2018; Jiang et al. 2018). Ren et al. (Ren et al. 2018) proposed a meta-learning method to reweight noisy data based on their gradient directions. Jiang et al. (Jiang et al. 2018) trained an auxiliary LSTM-based MentorNet to reweight noisy data. Both of these methods require an additional clean dataset, which can be costly in terms of labeling. Zhang et al. (Zhang et al. 2021b) presented a new meta re-weighting model without clean data, to reduce the labeling cost. Sun et al. (Sun et al. 2022a) proposed a hierarchical probabilistic method named warped probabilistic inference (WarPI) to reweight data, where they used a meta-network to estimate the distribution of noisy data. A boosting technique (Miao et al. 2015) is employed to update the weights using a manually designed re-weighting function. Another approach, introduced in the form of CleanNet (Lee et al. 2018), involves a joint neural embedding network aimed at minimizing the need for human supervision in label noise cleaning. CleanNet operates effectively with only a fraction of categories verified by human experts, and the knowledge acquired for label noise correction can subsequently be transferred to other classes. All of these methods usually assign small weights to noisy samples which reduce the contribution of noisy samples to the model and degenerate the robustness of DNN.

**Data filtering.** Another promising area focuses on training with clean data (Northcutt, Jiang, and Chuang 2021; Elkan and Noto 2008; Pleiss et al. 2020). These methods try to identify the noisy data, and then discard them before training the model. Xia et al. (Xia et al. 2021) addressed the uncertainty of losses by adopting interval estimation rather than point estimation. Specifically, they utilized the lower bounds of confidence intervals for losses, derived from distribution-free concentration inequalities, as criteria for data selection. Confident Learning (CL) (Northcutt, Jiang, and Chuang 2021) divided the training data into clean and noisy data then discarded the noisy data before training the model. $Elkan$ et al. (Elkan and Noto 2008) estimated noisy data using positive-




(a) SST-5-symmetric (b) SST-5-asymmetric (c) Amazon-5-symmetric (d) Amazon-5-asymmetric

Figure 5: Accuracy of SLaNT and DivideMix for identifying noisy data. (a) and (b): SST-5; (c) and (d): Amazon-5.




(a) SST-5-symmetric (b) SST-5-asymmetric (c) Amazon-5-symmetric (d) Amazon-5-asymmetric

Figure 6: Accuracy of relabeling strategy. (a) and (b): SST-5; (c) and (d): Amazon-5.

unlabeled learning which is limited to binary classification. AUM (Pleiss et al. 2020) measured the average difference between the logit values for each data assigned class to identify noisy data. However, all of them have a shortcoming. When the noise ratio is high, a significant amount of data would be discarded which reduces the robustness and generalization performance of DNNs. SELF (Nguyen et al. 2020) adopts the idea of self-ensemble to gradually identify the noisy labels and uses the Mean Teacher model (Tarvainen and Valpola 2017) to train DNNs with the entire dataset. However, SELF assumes the correct labels can be viewed as representative to achieve high model performance and no label correction is conducted, which hinders its application to sophisticated scenarios where the noisy labels are not randomly generated.

**Data relabeling.** These methods try to correct the noisy data and then train with clean data which leverages all the train data. Pnp (Sun et al. 2022b) introduced a straightforward yet highly effective approach where two networks are concurrently trained: one for predicting the category label, and the other for correcting the noise. Mallem et al. (Mallem, Hasnat, and Nakib 2023) first proposed a meta label correction net which needs a set of clean data to relabel noisy data. DivideMix (Li, Socher, and Hoi 2020) used the GMM to identify noisy data and trained two networks to relabel the noisy data. Unlike DivideMix (Li, Socher, and Hoi 2020), we only use multiple networks in the step of relabeling. The final re-training of our framework is merely performed on a single network, which makes SLaNT computationally much lighter. Moreover, both the strategies of noisy data identification and the relabeling in these works are different from SLaNT.

## Conclusion and Future Work

In this paper, our primary objective is to enhance the robustness of DNN models when confronted with the challenge of noisy labels in the context of sentiment analysis. To address this issue, we introduce a novel semi-supervised label noise-tolerant learning framework known as SLaNT. SLaNT can train a robust DNN model under noisy supervision through a pipeline of model early stopping, noisy data identification, data augmentation, noisy data relabeling and model re-training. Importantly, our extensive experimentation validates the superiority of SLaNT over current four state-of-the-art methods. In multiple instances, SLaNT achieves significantly improved results, enabling it to be an effective approach for addressing label noise in the domain of text sentiment analysis.

In future work, we are interested in exploring other effective components that can be embedded in SLaNT. We are also interested in adapting SLaNT to other social media scenarios, accommodating the specific characteristics and challenges of different domains.

## Acknowledgments

This research was partially supported by STI 2030-Major Projects 2021ZD0200400, National Natural Science Foundation of China (62276233 and 62072405) and Key Research Project of Zhejiang Province (2023C01048).

## References

Anomaly, U. T. S. 2022. Little Help Makes a Big Difference: Leveraging Active Learning to Improve Unsupervised Time Series Anomaly Detection. In *ICSOC*, volume 13236, 165. Springer Nature.

Arpit, D.; Jastrzebski, S.; Ballas, N.; Krueger, D.; Bengio, E.; Kanwal, M. S.; Maharaj, T.; Fischer, A.; Courville, A.; Bengio, Y.; et al. 2017. A closer look at memorization in deep networks. In *ICML*, 233–242. PMLR.

Bai, Y.; Yang, E.; Han, B.; Yang, Y.; Li, J.; Mao, Y.; Niu,

G.; and Liu, T. 2021. Understanding and Improving Early Stopping for Learning with Noisy Labels. *NIPS*, 34.

Bayer, M.; Kaufhold, M.-A.; Buchhold, B.; Keller, M.; Dallmeyer, J.; and Reuter, C. 2023. Data augmentation in natural language processing: a novel text generation approach for long and short text classifiers. *IJMLC*, 14(1): 135–150.

Berthelot, D.; Carlini, N.; Goodfellow, I.; Papernot, N.; Oliver, A.; and Raffel, C. A. 2019. MixMatch: A Holistic Approach to Semi-Supervised Learning. *NIPS*, 32: 5049–5059.

Brier, G. W.; et al. 1950. Verification of forecasts expressed in terms of probability. *Monthly weather review*, 78(1): 1–3.

Cireşan, D. C.; Meier, U.; Gambardella, L. M.; and Schmidhuber, J. 2010. Deep, big, simple neural nets for handwritten digit recognition. *Neural computation*, 22(12): 3207–3220.

Coulombe, C. 2018. Text data augmentation made simple by leveraging nlp cloud apis. *arXiv preprint arXiv:1812.04718*.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT*.

Du, J.; Grave, É.; Gunel, B.; Chaudhary, V.; Celebi, O.; Auli, M.; Stoyanov, V.; and Conneau, A. 2021. Self-training Improves Pre-training for Natural Language Understanding. In *NAACL-HLT*, 5408–5418.

Elkan, C.; and Noto, K. 2008. Learning classifiers from only positive and unlabeled data. In *KDD*, 213–220.

Feng, S. Y.; Gangal, V.; Wei, J.; Chandar, S.; Vosoughi, S.; Mitamura, T.; and Hovy, E. 2021. A Survey of Data Augmentation Approaches for NLP. In *ACL-IJCNLP*, 968–988.

Goldberger, J.; and Ben-Reuven, E. 2017. Training deep neural-networks using a noise adaptation layer. *ICLR*.

Gong, M.; Zhou, H.; Qin, A.; Liu, W.; and Zhao, Z. 2022. Self-Paced Co-Training of Graph Neural Networks for Semi-Supervised Node Classification. *TNNLS*.

Gururangan, S.; Marasović, A.; Swayamdipta, S.; Lo, K.; Beltagy, I.; Downey, D.; and Smith, N. A. 2020. Don't Stop Pretraining: Adapt Language Models to Domains and Tasks. In *ACL*, 8342–8360.

Han, B.; Yao, Q.; Yu, X.; Niu, G.; Xu, M.; Hu, W.; Tsang, I. W.; and Sugiyama, M. 2018. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *NIPS*.

Hou, Y.; Liu, Y.; Che, W.; and Liu, T. 2018. Sequence-to-Sequence Data Augmentation for Dialogue Language Understanding. In *ACL*, 1234–1245.

Jiang, L.; Zhou, Z.; Leung, T.; Li, L.-J.; and Fei-Fei, L. 2018. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *ICML*, 2304–2313. PMLR.

Kang, D.; Khot, T.; Sabharwal, A.; and Hovy, E. 2018. AdvEntuRe: Adversarial Training for Textual Entailment with Knowledge-Guided Examples. In *ACL*, 2418–2428.

Kearns, M. 1998. Efficient noise-tolerant learning from statistical queries. *JACM*, 45(6): 983–1006.

Lee, D.-M.; Kim, Y.; and gyun Seo, C. 2022. Context-based Virtual Adversarial Training for Text Classification with Noisy Labels. In *LREA*, 6139–6146.

Lee, K.-H.; He, X.; Zhang, L.; and Yang, L. 2018. Cleannet: Transfer learning for scalable image classifier training with label noise. In *CVPR*, 5447–5456.

Li, B.; Hou, Y.; and Che, W. 2021. Data Augmentation Approaches in Natural Language Processing: A Survey. *arXiv preprint arXiv:2110.01852*.

Li, J.; Socher, R.; and Hoi, S. C. 2020. DivideMix: Learning with Noisy Labels as Semi-supervised Learning. In *ICLR*.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Mallem, S.; Hasnat, A.; and Nakib, A. 2023. Efficient Meta label correction based on Meta Learning and bi-level optimization. *Engineering Applications of Artificial Intelligence*, 117: 105517.

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.

Miao, Q.; Cao, Y.; Xia, G.; Gong, M.; Liu, J.; and Song, J. 2015. RBoost: Label noise-robust boosting algorithm based on a nonconvex loss function and the numerically stable base learners. *TNNLS*, 27(11): 2216–2228.

Nguyen, T.; Mummadi, C.; Ngo, T.; Beggel, L.; and Brox, T. 2020. SELF: learning to filter noisy labels with self-ensembling. In *ICLR*.

Northcutt, C.; Jiang, L.; and Chuang, I. 2021. Confident learning: Estimating uncertainty in dataset labels. *JAIR*, 70: 1373–1411.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *NIPS*, 32: 8026–8037.

Pleiss, G.; Zhang, T.; Elenberg, E.; and Weinberger, K. Q. 2020. Identifying mislabeled data using the area under the margin ranking. *NIPS*, 33: 17044–17056.

Reed, S. E.; Lee, H.; Anguelov, D.; Szegedy, C.; Erhan, D.; and Rabinovich, A. 2015. Training Deep Neural Networks on Noisy Labels with Bootstrapping. In *ICLR*.

Ren, M.; Zeng, W.; Yang, B.; and Urtasun, R. 2018. Learning to reweight examples for robust deep learning. In *ICML*, 4334–4343. PMLR.

Shu, J.; Xie, Q.; Yi, L.; Zhao, Q.; Zhou, S.; Xu, Z.; and Meng, D. 2019. Meta-weight-net: Learning an explicit mapping for sample weighting. *NIPS*, 32.

Socher, R.; Perelygin, A.; Wu, J.; Chuang, J.; Manning, C. D.; Ng, A. Y.; and Potts, C. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *EMNLP*, 1631–1642.

Song, H.; Kim, M.; Park, D.; Shin, Y.; and Lee, J.-G. 2022. Learning from noisy labels with deep neural networks: A survey. *TNNLS*.

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

Sukhbaatar, S.; Bruna, J.; Paluri, M.; Bourdev, L.; and Fergus, R. 2015. Training convolutional networks with noisy labels. *ICLR*.

Sun, H.; Guo, C.; Wei, Q.; Han, Z.; and Yin, Y. 2022a. Learning to rectify for robust learning with noisy labels. *Pattern Recognition*, 124: 108467.

Sun, Z.; Shen, F.; Huang, D.; Wang, Q.; Shu, X.; Yao, Y.; and Tang, J. 2022b. Pnp: Robust learning from noisy labels by probabilistic noise prediction. In *CVPR*, 5311–5320.

Tanaka, D.; Ikami, D.; Yamasaki, T.; and Aizawa, K. 2018. Joint optimization framework for learning with noisy labels. In *CVPR*, 5552–5560.

Tarvainen, A.; and Valpola, H. 2017. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *ICONIP*, 1195–1204.

Tu, Y.; Zhang, B.; Li, Y.; Liu, L.; Li, J.; Wang, Y.; Wang, C.; and Zhao, C. R. 2023. Learning from noisy labels with decoupled meta label purifier. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 19934–19943.

Wei, J.; and Zou, K. 2019. EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks. In *EMNLP-IJCNLPE*, 6382–6388.

Wu, X.; Lv, S.; Zang, L.; Han, J.; and Hu, S. 2019. Conditional bert contextual augmentation. In *ICCS*, 84–95. Springer.

Wu, Y.; Shu, J.; Xie, Q.; Zhao, Q.; and Meng, D. 2021. Learning to purify noisy labels via meta soft label corrector. In *AAAI*, volume 35, 10388–10396.

Xia, X.; Liu, T.; Han, B.; Gong, M.; Yu, J.; Niu, G.; and Sugiyama, M. 2021. Sample Selection with Uncertainty of Losses for Learning with Noisy Labels. In *International Conference on Learning Representations*.

Yan, Y.; Xu, Z.; Tsang, I.; Long, G.; and Yang, Y. 2016. Robust semi-supervised learning through label aggregation. In *AAAI*, volume 30.

Yang, Z.; Dai, Z.; Yang, Y.; Carbonell, J.; Salakhutdinov, R. R.; and Le, Q. V. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *NIPS*, 32.

Zhang, C.; Bengio, S.; Hardt, M.; Recht, B.; and Vinyals, O. 2021a. Understanding deep learning (still) requires rethinking generalization. *COMMUN ACM*, 64(3): 107–115.

Zhang, H.; Cisse, M.; Dauphin, Y. N.; and Lopez-Paz, D. 2018. mixup: Beyond empirical risk minimization. *ICLR*.

Zhang, X.; Zhao, J.; and LeCun, Y. 2015. Character-level convolutional networks for text classification. *NIPS*, 28: 649–657.

Zhang, Y.; Liu, F.; Fang, Z.; Yuan, B.; Zhang, G.; and Lu, J. 2021b. Learning from a complementary-label source domain: theory and algorithms. *TNNLS*.

Zheng, G.; Awadallah, A. H.; and Dumais, S. 2021. Meta label correction for noisy label learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 11053–11061.

Zhou, T.; Wang, S.; and Bilmes, J. 2020. Robust curriculum learning: From clean label detection to noisy label self-correction. In *ICLR*.

## Paper Checklist

1. For most authors...
   (a) Would answering this research question advance science without violating social contracts, such as violating privacy norms, perpetuating unfair profiling, exacerbating the socio-economic divide, or implying disrespect to societies or cultures? Our method does not violate the social contract.
   (b) Do your main claims in the abstract and introduction accurately reflect the paper's contributions and scope? Yes.
   (c) Do you clarify how the proposed methodological approach is appropriate for the claims made? Yes.
   (d) Do you clarify what are possible artifacts in the data used, given population-specific distributions? Yes.
   (e) Did you describe the limitations of your work? No.
   (f) Did you discuss any potential negative societal impacts of your work? No.
   (g) Did you discuss any potential misuse of your work? No.
   (h) Did you describe steps taken to prevent or mitigate potential negative outcomes of the research, such as data and model documentation, data anonymization, responsible release, access control, and the reproducibility of findings? No.
   (i) Have you read the ethics review guidelines and ensured that your paper conforms to them? Yes.
2. Additionally, if your study involves hypotheses testing...
   (a) Did you clearly state the assumptions underlying all theoretical results? N/A.
   (b) Have you provided justifications for all theoretical results? Answer
   (c) Did you discuss competing hypotheses or theories that might challenge or complement your theoretical results? N/A.
   (d) Have you considered alternative mechanisms or explanations that might account for the same outcomes observed in your study? N/A.
   (e) Did you address potential biases or limitations in your theoretical framework? N/A.
   (f) Have you related your theoretical results to the existing literature in social science? N/A.
   (g) Did you discuss the implications of your theoretical results for policy, practice, or further research in the social science domain? N/A.
3. Additionally, if you are including theoretical proofs...
   (a) Did you state the full set of assumptions of all theoretical results? N/A.
   (b) Did you include complete proofs of all theoretical results? N/A.

4. Additionally, if you ran machine learning experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? No. We will release the code after the paper is officially accepted
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? Yes.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? Yes.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? Yes.
   (e) Do you justify how the proposed evaluation is sufficient and appropriate to the claims made? Yes.
   (f) Do you discuss what is "the cost" of misclassification and fault (in)tolerance? No.
5. Additionally, if you are using existing assets (e.g., code, data, models) or curating/releasing new assets, **without compromising anonymity**...
   (a) If your work uses existing assets, did you cite the creators? Yes.
   (b) Did you mention the license of the assets? No.
   (c) Did you include any new assets in the supplemental material or as a URL? No.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? No. In this paper, the datasets are all public.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? No.
   (f) If you are curating or releasing new datasets, did you discuss how you intend to make your datasets FAIR (see **?**)? N/A.
   (g) If you are curating or releasing new datasets, did you create a Datasheet for the Dataset (see **?**)? N/A.
6. Additionally, if you used crowdsourcing or conducted research with human subjects, **without compromising anonymity**...
   (a) Did you include the full text of instructions given to participants and screenshots? N/A.
   (b) Did you describe any potential participant risks, with mentions of Institutional Review Board (IRB) approvals? N/A.
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? N/A.
   (d) Did you discuss how data is stored, shared, and deidentified? N/A.