SaFER: A Robust and Efficient Framework for Fine-tuning BERT-based Classifier with Noisy Labels

Zhenting Qi[*] [1] **Xiaoyu Tan**[*] [2†] **Chao Qu**[2] **Yinghui Xu**[3] **Yuan Qi**[3]
[1]Zhejiang University [2] INF Technology (Shanghai) Co., Ltd. [3] Fudan University
Zhenting.19@intl.zju.edu.cn, yulin.txy@inftech.ai

Abstract

Learning on noisy datasets is a challenging problem when pre-trained language models are applied to real-world text classification tasks. In numerous industrial applications, acquiring task-specific datasets with 100% accurate labels is difficult, thus many datasets are accompanied by label noise at different levels. Previous work has shown that existing noise-handling methods could not improve the peak performance of BERT on noisy datasets, and might even deteriorate it. In this paper, we propose SaFER[1], a robust and efficient fine-tuning framework for BERT-based text classifiers, combating label noises without access to any clean data for training or validation. Utilizing a label-agnostic early-stopping strategy and self-supervised learning, our proposed framework achieves superior performance in terms of both accuracy and speed on multiple text classification benchmarks. The trained model is finally fully deployed in several industrial biomedical literature mining tasks and demonstrates high effectiveness and efficiency.

1 Introduction

Large Language Models (LLMs) have dominated Natural Language Processing (NLP) in recent years and achieved state-of-the-art performance in a variety of industrial applications. Among them, the most widely-adapted LLMs are transformer-based models, including BERT, T5, GPT, etc (Devlin et al., 2018; Raffel et al., 2020; Brown et al., 2020). LLMs learn general natural language knowledge from large corpora and the representations for text data can be utilized in various downstream NLP tasks. Such a paradigm is also extensively implemented in both industrial and research domains and achieves considerable performance improvement compared with traditional statistical approaches.

Text classification is one of the most important tasks in the industrial domain (Sanchez-Pi et al., 2014; Han and Akbari, 2018; Wei et al., 2018; Arslan et al., 2021; Chen et al., 2018; Cheng et al., 2021), including sentence classification, named entity recognition, etc. Typically, these tasks can be accomplished by leveraging embeddings generated by the encoder architecture of LLMs. Unfortunately, the performance is always limited by the data quality in either pre-training or fine-tuning stage. Real-world datasets, especially those collected for industrial applications, usually contain a substantial proportion of mislabeled data (Song et al., 2022). Such label noise can be induced by crowd-sourcing, human mistakes, system errors, or the uncertainty itself in the weakly-supervised labeling methodology. The corrupted labels can dramatically influence the model performance and robustness, which has been validated theoretically and experimentally (Song et al., 2022; Zhu et al., 2022b). Worse still, re-labeling procedures can be cost-intensive and time-consuming due to lack of domain experts. That means in most cases we can only access the noisy validation set and lose the validation with ground truth.

Previous work addressed the label noise issue by proposing robust loss functions, recovering the noise transition matrix, and incorporating unsupervised learning strategies (Jindal et al., 2019; Yao et al., 2020; Lukasik et al., 2020; Jenni and Favaro, 2018; Wei et al., 2020; Tan et al., 2021). However, the label noise issue of using LLMs in NLP tasks is still an open problem and remains unsolved, especially in text classification tasks. Zhu et al. (2022b) demonstrate that directly incorporating existing noise-encountering methods cannot consistently improve and even deteriorate the BERT model performance under noisy labels in text classification. This conclusion is also supported by the results of our industrial implementation and ablation study (Appendix C). By investigating the process of fine-

[*]Equal Contributions.
[†]Corresponding author.
[1]Code will be released at GitHub.

tuning LLMs on noisy sets, we observe that the model is experiencing a “convex” learning curve under label noise (Appendix C Figure 3): the model gradually increases the accuracy by learning easy samples in the earlier stage, but continuously experiences performance drop by over-fitting the noise labels. Hence, *can we reserve the knowledge of the first stage and mitigate the over-fitting in the second stage without using any clean data validation?*

Following this intuition, we propose SaFER: a noi**S**e-resist**a**nt **F**ramework for **E**fficient and **R**obust BERT fine-tuning to perform robust LLMs fine-tuning under noisy classification NLP tasks, *without using any clean labels*. Generally, this framework is compatible with any LLM that contains an encoder architecture to encode sequences into latent representations. We first fine-tune the model following a typical manner but perform early stopping with a label-agnostic strategy. Then, we leverage contrastive learning with an NLP-specific augmentation strategy and implement structural learning to further combat noisy labels.

To evaluate our proposed framework, we perform experiments on text classification tasks using pre-trained BERT models (Devlin et al., 2018). Here, we select the BERT family to represent LLMs due to their widespread recognition in both the industry and research domains. We implement several state-of-the-art robust learning methods against label noise as our baseline methods (Appendix B.1). These methods are designed to mitigate the general classification label noise issue without explicitly considering the usage of LLMs. The experiment results show the consistent and superior performance of our proposed learning framework. Finally, we implement SaFER in two industrial biomedical literature mining tasks under unavoidable human labeling noise and achieve robust practical performance compared with baselines.

The main contributions are as follows:

- We propose an efficient and robust learning framework: SaFER, for BERT fine-tuning on datasets with noisy labels without accessing any clean data.
- We empirically show that SaFER achieves superior performance on text classification tasks using BERT.
- We demonstrate the practical feasibility of SaFER on two industrial biomedical literature mining tasks.

2 Methods

2.1 Problem Settings

We focus on the classification task on text data. Suppose $X \subset \mathbb{R}^d$ is the d dimensional input space and $Y = \{0, 1\}^k$ is the label space in a one-hot manner. In a typical text classification task, a clean training corpus $C = \{(x_i, y_i)_{i=1}^n\}$ drawn from the joint distribution $X \times Y$ is provided, where x_i denotes a data sample, y_i is the corresponding ground-truth label of x_i, and n is the size of the corpus. However, in the noisy label setting, a certain proportion of the training data are not correctly labeled. Given a noisy training corpus $\widehat{C} = \{(x_i, \hat{y}_i)_{i=1}^n\}$ drawn from a noisy joint distribution $X \times \hat{Y}$, with noise level being $\rho = |\{(x_i, \hat{y}_i)|\hat{y}_i \neq y_i\}|/n$, we hope to train a classifier $f(\cdot;\theta)$ that gives correct predictions on unseen data.

Some noise handling methods assume that a small clean set is available (Tänzer et al., 2021; Shu et al., 2019). However, such clean data is often not easy to obtain in real-world industrial settings. In our problem, we assume that there is no fully clean data available. In the subsequent sections, by saying “noisy” we mean there is a non-zero probability that such data item is wrongly labeled.

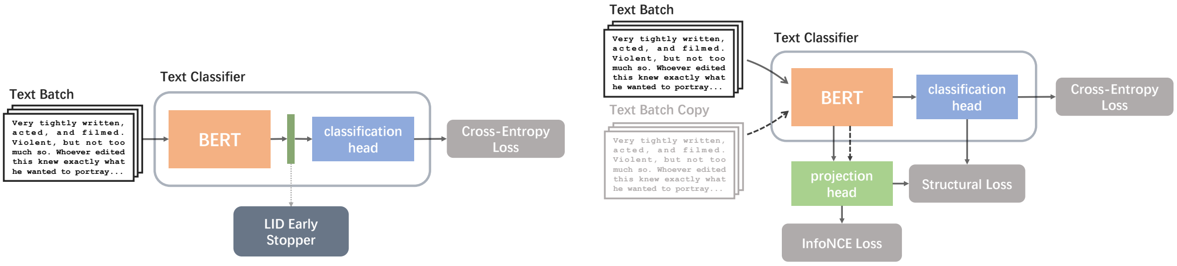
2.2 SaFER

2.2.1 Motivation

We identify two problems caused by noisy labels: 1) early stopping at the wrong training step on noisy validation sets, therefore one would miss the best model parameters, and 2) noisy supervision from incorrect labels, thus preventing the model from improving its performance or even deteriorating it.

Improper Early Stopping Traditional strategy (Tänzer et al., 2021) relies on a clean validation set to find the point where the model reaches its highest generalization capability. If a model reaches a high performance on the validation set at some training step, and such performance is not exceeded in a certain amount of further steps, then the model is early stopped. In our settings, nevertheless, such a validation set is not clean and the performance on such a set may not be a reliable indicator of early stopping, as shown in Appendix C Figure 4. Moreover, evaluating the model for every few steps is very time-consuming, especially when it comes to doing inference with large language models with a huge amount of parameters.

Noisy Supervision Under label noises, the optimization with loss function $\mathbf{L}$ on a noisy batch $\widehat{C}_B$


Text Batch
Very tightly written, acted, and filmed. Violent, but not too much so. Whoever edited this knew exactly what he wanted to portray...
Text Classifier
BERT
classification head
Cross-Entropy Loss
LID Early Stopper
Text Batch
Very tightly written, acted, and filmed. Violent, but not too much so. Whoever edited this knew exactly what he wanted to portray...
Text Batch Copy
Very tightly written, acted, and filmed. Violent, but not too much so. Whoever edited this knew exactly what he wanted to portray...
Text Classifier
BERT
classification head
Cross-Entropy Loss
projection head
Structural Loss
InfoNCE Loss

(a) **STAGE 1**: Fast Warming Up

(b) **STAGE 2**: Combating Label Noises

Figure 1: Illustration of SaFER framework.

sampled from the noisy dataset $\widehat{C}$ at time step t is formulated as:

$$\theta_{t+1} = \theta_t - \eta \nabla \left(\frac{1}{|\widehat{C}_B|} \sum_{(x,\hat{y}) \in \widehat{C}_B} \mathbf{L}(f(x;\theta_t), \hat{y}) \right), \quad (1)$$

where the noisy label $\hat{y}$ participates in the loss calculation and can corrupt the model parameters through backward propagation, thus misleading the optimization progress.

2.2.2 Label-Independent Early Stopping

To find the point where the generalization reaches the best performance, we need a reliable signal that indicates when the classifier starts to overfit the noisy labels. Intrinsic Dimensionality (ID) is a measure of the number of variables needed to minimally represent a set of data, and it has been proven to be a good indicator of the generalization ability of DNNs (Amsaleg et al., 2017; Ma et al., 2018b,a; Ansuini et al., 2019; Nakada and Imaizumi, 2020; Birdal et al., 2021). When considering each sample in the dataset, we have Local Intrinsic Dimensionality (LID) that measures the dimensional complexity of the local subspace in its vicinity. Previous studies on DNN learning dynamics empirically show that the ID curve behaves like a concave shape at the beginning of the training, which applies to several types of DNNs. When LID reaches a low point, the DNN could reach high generalization ability before starting to memorize label noises. We show that the BERT encoder also follows such a manner (Appendix C Figure 2) and propose to utilize such characteristics to find the proper stopping point.

Following Ma et al. (2018b), we estimate the LID score within batches. Consider a BERT-based classifier $h : P \to \mathbb{R}^k$ where k is the number of classes. Given a transformed text batch $X_B \subset X$ sampled from the training corpus, and a reference point $x \sim P$, the estimated LID score of x can be calculated as:

$$\widehat{\mathrm{LID}}(x, X_B) = - \left(\frac{1}{k} \sum_{i=1}^{k} \log \frac{r_i(g(x), g(X_B))}{r_{max}(g(x), g(X_B))} \right)^{-1}, \quad (2)$$

where g is the output from the second-to-last layer of BERT, $r_i(g(x), g(X_B))$ is the distance of $g(x)$ to its i-th nearest neighbor in the batch, and $r_{max}(g(x), g(X_B))$ is the maximum distance of $g(x)$ to its neighbors in the batch (or the radius of the batch, centered on x). Using the estimated LID score as a stopping indicator, we can quickly warm up the classifier without any noise-handling modules, as shown in stage 1 of Algorithm 1 in Appendix D. As for why we do not apply any noise handling at the first stage, Zhu et al. (2022b) has shown in Figure 1 of their paper that the pure BERT method is the fastest one to reach its peak performance compared with other methods that add extra modules to BERT, and such peak performance is already very high, thus being a good start.

2.2.3 Noise-Tolerant Supervisor

To further improve the model's performance after warming up, we introduce unsupervised learning to apply label-independent supervision, thus preventing label noises from misleading the optimization. The second stage incorporates a simple multi-layer perceptron (MLP) as a projection head, which is trained along with the classifier and simultaneously imposes constraints on the representation space of the classifier.

Given a noisy batch $\widehat{C}_B = \{(x_i, \hat{y}_i)_{i=1}^{b}\}$ drawn from $\widehat{C}$, the BERT encoder f with parameters θ_1 takes the tokenized sentence x_i as input and produces the sentence embedding $z_i = f(x_i; \theta_1)$. The classification head g with parameters θ_2 transforms z_i into a k-dimensional vector $u_i = g(z_i; \theta_2)$ (k is the number of classes), which is supervised by noisy labels with cross-entropy loss:

$$\mathbf{L}_{CE}(u_i, \hat{y_i}) = -\sum_{j=1}^{k} \hat{y_i}^{(j)} \log(u_i^{(j)}),$$
$$\mathcal{L}_{CE}(\{u_i\}_{i=1}^{b}, \{\hat{y_i}\}_{i=1}^{b}) = \sum_{i=1}^{b} \mathbf{L}_{CE}(u_i, \hat{y_i}). \tag{3}$$

On the other hand, we train a projection head h with parameters θ_3 using contrastive learning. Borrowing experience from the unsupervised SimCSE framework (Gao et al., 2021), we forward passed x_i through f and g twice and got two representations $v_i = h(f(x_i, m; \theta_1); \theta_3)$ and $v_i' = h(f(x_i, m'; \theta_1); \theta_3)$ as positive pairs, where m and m' are two random dropout masks of BERT, while representations for other sentences are treated as negative samples. The reason why we adopted SimCSE is that it is an extremely simple but efficient way to build positive pairs for sentences, and it has been proven to perform much better in sentence representation than other traditional NLP data augmentations such as crop and word deletion. Using the output representations, we calculate the standard InfoNCE loss (Oord et al., 2018):

$$\mathbf{L}_{con}(v_i) = -\log \frac{\exp(S(v_i, v_i')/\tau)}{\sum_{j=1}^{b} \exp(S(v_i, v_j')/\tau)},$$
$$\mathcal{L}_{con}(\{v_i\}_{i=1}^{b}) = \sum_{i=1}^{b} \mathbf{L}_{con}(v_i), \tag{4}$$

where S is a measurement of similarity between representation vectors and τ is a temperature hyperparameter. As for why we need such a projection head h, the reason is that it prevents noisy labels from corrupting the classifier's representation space by forcing the classification head g to "agree" with its output. This is realized by applying a structural loss (Tan et al., 2021) to h and g. Minimizing the KL-divergence between the similarities of classifier outputs and those of projector outputs, structural loss applies a structure-preserving constraint on output features of g, keeping its representation space structure similar to that of h:

$$\mathcal{L}_{str}(\{u_i\}_{i=1}^{b}, \{v_i\}_{i=1}^{b}) = \sum_{p \neq q} R(v_p, v_q) \log \frac{R(v_p, v_q)}{R(u_p, u_q)}, \tag{5}$$

where R is a similarity metric. Notice that only the classification head is trained with structural loss, while BERT and the projection head are frozen.

2.2.4 Two-Stage Fine-tuning Framework

We design a two-stage framework for fine-tuning models. For the first stage, a classifier with pretrained BERT as the backbone is fine-tuned on the noisy set without any noise-handling methods, but monitored by a reliable early stopper. After the early stopping is triggered, we enter the second stage where a projection head is trained along with the classifier using an unsupervised learning method, and it simultaneously applies regularization to the representation space of the classifier. With a strong knowledge base built at the first stage and further boosting at the second stage, the classifier can reach a high generalization capability quickly. The entire SaFER framework is illustrated in Figure 1 and Algorithm 1 in Appendix D.

3 Experiment

3.1 Implementation

Injected Label Noise Following previous work on modeling noisy datasets (Reed et al., 2014; Van Rooyen et al., 2015), we define two types of synthetic label noise: the single-flip noise

$$\mathcal{P}(\hat{y} = j | y = i) = \begin{cases} 1-\rho, & \text{for } i = j \\ \rho, & \text{for one } i \neq j \\ 0, & \text{else} \end{cases}$$

and the uniform-flip noise

$$\mathcal{P}(\hat{y} = j | y = i) = \begin{cases} 1-\rho, & \text{for } i = j \\ \dfrac{\rho}{k-1}, & \text{else} \end{cases}$$

According to statistics shown in the survey done by Song et al. (2022), we define four levels of injected noises: low ($\rho = 0.2$), medium ($\rho = 0.3$), high ($\rho = 0.4$), and extreme ($\rho = 0.45$).

Models We use the pre-trained BERT-base model from Huggingface as the pre-trained BERT backbone, and the BERT fine-tuning strategy follows Devlin et al. (2018). The classifier head is a linear layer with input size being the hidden size of the BERT backbone and output size being the number of classes. The projection head is a two-layer perceptron in which the input size is the hidden size of the BERT backbone, the intermediate size is 512, and the output size is 128 (i.e., the feature dimension of the projection).

Baselines We compare SaFER with the following methods (implementations were adapted from code provided by Zhu et al. (2022b)): Without Noise-Handling, Noise Matrix, Noise Matrix with Regularization, Label Smoothing, Robust Loss, Co-Teaching, and Co-Learning. Note that all of them use a noisy validation set for early stopping. We refer readers to Appendix B.1 for more details.

Environment The model is fine-tuned by single NVIDIA Tesla V100-32G GPU under PyTorch (v1.12.1) framework. We refer the readers to Appendix B.2 for more implementation details.

3.2 Text Classification with BERT

IMDB (Maas et al., 2011) is a dataset for binary sentiment classification containing around 50K movie reviews, most commonly used for sentiment analysis, i.e. models should predict "positive" or "negative" for the reviews. We use a set of 25K reviews for training/validation, and 25K for testing. Following Zhu et al. (2022b), we inject single-flip noise into the IMDB dataset.

AG-News (Zhang et al., 2015) is a sub-dataset of AG's corpus of more than 1 million news articles gathered from more than 2K news sources, having the 4 largest classes ("World", "Sports", "Business", "Sci/Tech") of AG's Corpus. The AG-News contains 30K training samples and 1,900 test samples for each class. Following Zhu et al. (2022b), we inject uniform-flip noise into the AG-News dataset.

Before injecting label noises, we assume that IMDB and AG-News themselves are 100% clean. Their test splits remain the same, while the training/validation splits are modified by the aforementioned injected noises. Table 1 and 2 show the experiment results: SaFER performs the best across all noise levels on AG-News, and also achieves state-of-the-art performance on IMDB, except that at medium noise level, it reaches comparable results with Co-Learning. When the noise level is low or medium, all methods maintain good performance. But when it comes to higher noise, all the baselines showed varying degrees of decline in accuracy, while SaFER still maintains high performance thanks to the feature-dependent information gained from unsupervised learning. Especially, we notice that on extremely noisy IMDB, Co-Teaching stops at the wrong training step where the accuracy is just 52.53% because the validation set is so noisy that the highest validation accuracy does not match the highest test accuracy (above 70%). But SaFER uses label-independent method for early stopping, thus avoiding such a problem. Notably, SaFER is the only method that maintains accuracy above 90% on AG-News across all five levels of noise.

Regarding efficiency, SaFER has also shown superior results compared with other baseline methods. Noise Matrix, Label Smoothing, and Robust Loss do not differ much from pure BERT because they introduce only limited extra computations. But Co-Teaching needs to maintain two neural networks, thus being very slow during backward propagation, and Co-Learning trains the BERT backbone twice: once with the classifier and once with the projector, therefore it is also very inefficient. However, SaFER uses pure BERT for the first stage, which largely cut down the training time. Most importantly, the LID-based early stopping strategy does not require inference on a validation set, thus saving much time at each evaluation step. As shown in the table, SaFER only takes around half of the time per training step that is required by Co-Teaching and Co-Learning.

3.3 Ablation Study

To verify the effectiveness of using two-stage, we compare SaFER with one-stage pure BERT and one-stage BERT with unsupervised learning. The test accuracy v.s. training step on IMDB is shown in Appendix C Figure 3. Note that the two-stage scheme actually uses pure BERT for stage one and next uses BERT with unsupervised learning for stage two. As we can see, pure BERT climbs up very fast at the initial 500 steps, while BERT with unsupervised learning cannot reach the same accuracy until its 1500th step. However, pure BERT's accuracy starts to drop after it reaches maximum performance, while BERT with unsupervised learning continues going up. SaFER combines their advantages: the accuracy quickly gets to a high point and continues climbing with a stable pace, therefore its curve is at the highest place.

Early stopping for stage one is used to find the transition point from pure BERT to BERT with unsupervised learning. We also studied whether to apply early stopping to stage two to find the converging point of the classifier, as shown in Appendix C Table 4. It can be seen that with early stopping at stage 2, SaFER reaches an accuracy of **89.06%**, which is much higher than pure BERT's

Methods	$\rho = 0.0$	$\rho = 0.2$	$\rho = 0.3$	$\rho = 0.4$	$\rho = 0.45$	time/step
Without Noise-Handling	93.36	91.08	90.96	86.72	77.76	5.43s
Noise Matrix	93.18	91.19	90.30	87.37	81.45	5.49s
Noise Matrix with Regularization	93.29	91.40	90.71	88.16	79.26	5.42s
Label Smoothing	93.34	91.49	90.10	86.42	73.72	5.48s
Robust Loss: MAE	91.98	88.74	85.98	78.66	73.93	5.46s
Robust Loss: SCE	93.26	88.75	85.74	83.92	76.21	5.89s
Co-Teaching	93.42	90.98	90.96	84.84	52.53	5.94s
Co-Learning	93.56	91.83	**91.46**	86.76	78.37	6.15s
SaFER	**93.73**	**92.64**	91.27	**89.06**	**82.48**	**3.31s**

Table 1: Comparing accuracy(%) with SOTA methods on **IMDB**. ρ stands for noise level, and time/step is the average time needed for each training step (including the necessary time for validation or LID score calculation).

Methods	$\rho = 0.0$	$\rho = 0.2$	$\rho = 0.3$	$\rho = 0.4$	$\rho = 0.45$	time/step
Without Noise-Handling	91.35	90.76	88.63	85.32	87.34	3.82s
Noise Matrix	92.93	89.00	88.67	85.34	83.27	3.97s
Noise Matrix with Regularization	90.72	89.23	88.52	85.73	84.60	3.84s
Label Smoothing	91.35	89.97	89.72	90.12	88.53	3.90s
Robust Loss: MAE	90.18	90.01	90.12	89.03	87.89	3.92s
Robust Loss: SCE	92.98	90.13	88.71	89.38	88.80	4.03s
Co-Teaching	91.25	89.89	87.84	87.02	86.37	4.40s
Co-Learning	91.82	90.78	89.82	89.86	88.30	4.52s
SaFER	**93.07**	**92.22**	**91.66**	**91.13**	**90.92**	**2.16s**

Table 2: Comparing accuracy(%) with SOTA methods on **AG-News**. ρ stands for noise level, and time/step is the average time needed for each training step (including the necessary time for validation or LID score calculation).

84.49% and BERT with unsupervised learning's 77.33% and even better than fine-tuning BERT with unsupervised learning till the end (88.59%).

3.4 Biomedical Literature Mining

We further deploy our framework on two industrial biomedical literature mining tasks. These tasks are binary classification tasks used to recognize special biomedical phrases in the literature to assist our medication and biomedical experts in patents and literature reading. The data is acquired from several experts in daily work who have different technology stacks. The data is labeled by experts themselves or organized from the web resource in daily work. Hence, the data itself is highly corrupted by label noise due to crowd-sourcing and labeling preference. Unfortunately, unifying the label standard and relabeling all data is impossible due to the high workload of our experts and the large quantity of data: both tasks share the same data space which has around 40K data with an average of 60 text lengths. To evaluate our proposed method, we invite one human expert to examine and relabel part of the dataset which contains 2K data, and suppose that this part of the data is clean. We use this part of the data as a test set for model evaluation and find that the label noise level for both tasks is around $\rho = 0.3$. We shuffle and split the remaining noisy data by 20% and 80% for validation and training, and fine-tune the original BERT model in a typical training manner and our proposed SaFER framework, separately. Experiment results are listed in Table 3, showing the practical effectiveness of SaFER in industrial settings. We deploy our trained model as a new online service in our company to assist biomedical researchers in literary readings.

Methods	Task 1	Task 2
BERT w/o noise-handling	75.24	91.02
SaFER	**80.03**	**94.75**

Table 3: Accuracy (%) for two industrial biomedical literature mining tasks.

4 Conclusion

We propose a novel framework SaFER to perform robust and efficient BERT fine-tuning in text classification tasks under label noises. This framework is evaluated on both open-source datasets with synthetic label noise and industrial tasks with human label noise, compared with several state-of-the-art noise handling methods. Experiments show that SaFER not only achieves superior results but also demonstrates significant improvement in efficiency.

Limitations

SaFER framework is designed for handling BERT classification label noise without using any clean data. Despite the fact that the BERT is one of the most extensively used models in the industrial domain, the influence of label noise on GPT models and prompt should be further studied in light of the recent rapid progress. We believe that our framework is compatible with these models, however, further evaluation is required.

Another limitation is the types of label noise. We analyze SaFER using synthetic datasets with uniform and flip label noise which are typical class-level noise in practice. However, in industrial applications, the model may experience instance-level label noise, which is beyond the scope of our investigation. Although SaFER achieves robust results in our biomedical literature mining task under human label noise, we encourage users to examine the label noise type first in their own application.

Ethics Statement

All experiments can be conducted on a single NVIDIA Tesla V100-32G GPU. The datasets (Maas et al., 2011; Zhang et al., 2015) used to compare SaFER with previous methods are publicly available, and we did not modify any data or labels in these datasets. The dataset used for industrial biomedical literature mining tasks is protected and we do not plan to make it public in this work. But the source code and instructions for using our framework will be released along with the paper.

References

Laurent Amsaleg, James Bailey, Dominique Barbe, Sarah Erfani, Michael E Houle, Vinh Nguyen, and Miloš Radovanović. 2017. The vulnerability of learning to adversarial perturbation increases with intrinsic dimensionality. In *2017 ieee workshop on information forensics and security (wifs)*, pages 1–6. IEEE.

Alessio Ansuini, Alessandro Laio, Jakob H Macke, and Davide Zoccolan. 2019. Intrinsic dimension of data representations in deep neural networks. *Advances in Neural Information Processing Systems*, 32.

Yusuf Arslan, Kevin Allix, Lisa Veiber, Cedric Lothritz, Tegawendé F Bissyandé, Jacques Klein, and Anne Goujon. 2021. A comparison of pre-trained language models for multi-class text classification in the financial domain. In *Companion Proceedings of the Web Conference 2021*, pages 260–268.

Alan Joseph Bekker and Jacob Goldberger. 2016. Training deep neural-networks based on unreliable labels. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2682–2686. IEEE.

Tolga Birdal, Aaron Lou, Leonidas J Guibas, and Umut Simsekli. 2021. Intrinsic dimension, persistent homology and generalization in neural networks. *Advances in Neural Information Processing Systems*, 34:6776–6789.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Lingzhen Chen, Alessandro Moschitti, Giuseppe Castellucci, Andrea Favalli, and Raniero Romagnoli. 2018. Transfer learning for industrial applications of named entity recognition. In *NL4AI@ AI* IA*, pages 129–140.

Xiang Cheng, Mitchell Bowden, Bhushan Ramesh Bhange, Priyanka Goyal, Thomas Packer, and Faizan Javed. 2021. An end-to-end solution for named entity recognition in ecommerce search. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 15098–15106.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821*.

Aritra Ghosh, Himanshu Kumar, and P Shanti Sastry. 2017. Robust loss functions under label noise for deep neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31.

Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. 2018. Co-teaching: Robust training of deep neural networks with extremely noisy labels. *Advances in neural information processing systems*, 31.

Jianglei Han and Mohammad Akbari. 2018. Vertical domain text classification: towards understanding it tickets using deep neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Dan Hendrycks, Mantas Mazeika, Duncan Wilson, and Kevin Gimpel. 2018. Using trusted data to train deep networks on labels corrupted by severe noise. *Advances in neural information processing systems*, 31.

Simon Jenni and Paolo Favaro. 2018. Deep bilevel learning. In *Proceedings of the European conference on computer vision (ECCV)*, pages 618–633.

Ishan Jindal, Daniel Pressel, Brian Lester, and Matthew Nokleby. 2019. An effective label noise model for dnn text classification. *arXiv preprint arXiv:1903.07507*.

Himanshu Kumar, Naresh Manwani, and PS Sastry. 2020. Robust learning of multi-label classifiers under label noise. In *Proceedings of the 7th ACM IKDD CoDS and 25th COMAD*, pages 90–97.

Michal Lukasik, Srinadh Bhojanapalli, Aditya Menon, and Sanjiv Kumar. 2020. Does label smoothing mitigate label noise? In *International Conference on Machine Learning*, pages 6448–6458. PMLR.

Xingjun Ma, Bo Li, Yisen Wang, Sarah M Erfani, Sudanthi Wijewickrema, Grant Schoenebeck, Dawn Song, Michael E Houle, and James Bailey. 2018a. Characterizing adversarial subspaces using local intrinsic dimensionality. *arXiv preprint arXiv:1801.02613*.

Xingjun Ma, Yisen Wang, Michael E Houle, Shuo Zhou, Sarah Erfani, Shutao Xia, Sudanthi Wijewickrema, and James Bailey. 2018b. Dimensionality-driven learning with noisy labels. In *International Conference on Machine Learning*, pages 3355–3364. PMLR.

Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pages 142–150.

Eran Malach and Shai Shalev-Shwartz. 2017. Decoupling" when to update" from" how to update". *Advances in neural information processing systems*, 30.

Milad Moradi and Matthias Samwald. 2021. Evaluating the robustness of neural language models to input perturbations. *arXiv preprint arXiv:2108.12237*.

Ryumei Nakada and Masaaki Imaizumi. 2020. Adaptive approximation and generalization of deep neural network with intrinsic dimensionality. *The Journal of Machine Learning Research*, 21(1):7018–7055.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. 2017. Making deep neural networks robust to label noise: A loss correction approach. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1944–1952.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Scott Reed, Honglak Lee, Dragomir Anguelov, Christian Szegedy, Dumitru Erhan, and Andrew Rabinovich. 2014. Training deep neural networks on noisy labels with bootstrapping. *arXiv preprint arXiv:1412.6596*.

Nayat Sanchez-Pi, Luis Martí, and Ana Cristina Bicharra Garcia. 2014. Text classification techniques in oil industry applications. In *International Joint Conference SOCO'13-CISIS'13-ICEUTE'13: Salamanca, Spain, September 11th-13th, 2013 Proceedings*, pages 211–220. Springer.

Jun Shu, Qi Xie, Lixuan Yi, Qian Zhao, Sanping Zhou, Zongben Xu, and Deyu Meng. 2019. Meta-weight-net: Learning an explicit mapping for sample weighting. *Advances in neural information processing systems*, 32.

Hwanjun Song, Minseok Kim, Dongmin Park, Yooju Shin, and Jae-Gil Lee. 2022. Learning from noisy labels with deep neural networks: A survey. *IEEE Transactions on Neural Networks and Learning Systems*.

Sainbayar Sukhbaatar, Joan Bruna, Manohar Paluri, Lubomir Bourdev, and Rob Fergus. 2014. Training convolutional networks with noisy labels. *arXiv preprint arXiv:1406.2080*.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826.

Cheng Tan, Jun Xia, Lirong Wu, and Stan Z Li. 2021. Co-learning: Learning from noisy labels with self-supervision. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 1405–1413.

Michael Tänzer, Sebastian Ruder, and Marek Rei. 2021. Memorisation versus generalisation in pre-trained language models. *arXiv preprint arXiv:2105.00828*.

Brendan Van Rooyen, Aditya Menon, and Robert C Williamson. 2015. Learning with symmetric label noise: The importance of being unhinged. *Advances in neural information processing systems*, 28.

Boxin Wang, Shuohang Wang, Yu Cheng, Zhe Gan, Ruoxi Jia, Bo Li, and Jingjing Liu. 2020. Infobert: Improving robustness of language models from an information theoretic perspective. *arXiv preprint arXiv:2010.02329*.

Yisen Wang, Xingjun Ma, Zaiyi Chen, Yuan Luo, Jinfeng Yi, and James Bailey. 2019. Symmetric cross entropy for robust learning with noisy labels. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 322–330.

Fusheng Wei, Han Qin, Shi Ye, and Haozhen Zhao. 2018. Empirical study of deep learning for text classification in legal document review. In *2018 IEEE*

International Conference on Big Data (Big Data), pages 3317–3320. IEEE.

Hongxin Wei, Lei Feng, Xiangyu Chen, and Bo An. 2020. Combating noisy labels by agreement: A joint training method with co-regularization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13726–13735.

Yu Yao, Tongliang Liu, Bo Han, Mingming Gong, Jiankang Deng, Gang Niu, and Masashi Sugiyama. 2020. Dual t: Reducing estimation error for transition matrix in label-noise learning. *Advances in neural information processing systems*, 33:7260–7271.

Xingrui Yu, Bo Han, Jiangchao Yao, Gang Niu, Ivor Tsang, and Masashi Sugiyama. 2019. How does disagreement help generalization against label corruption? In *International Conference on Machine Learning*, pages 7164–7173. PMLR.

Shengnan Zhang, Yuexian Hou, Benyou Wang, and Dawei Song. 2017. Regularizing neural networks via retaining confident connections. *Entropy*, 19(7):313.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28.

Bin Zhu, Zhaoquan Gu, Le Wang, Jinyin Chen, and Qi Xuan. 2022a. Improving robustness of language models from a geometry-aware perspective. *arXiv preprint arXiv:2204.13309*.

Dawei Zhu, Michael A Hedderich, Fangzhou Zhai, David Ifeoluwa Adelani, and Dietrich Klakow. 2022b. Is bert robust to label noise? a study on learning with noisy labels in text classification. *arXiv preprint arXiv:2204.09371*.

A Related Work

In this section, we briefly review previous work on the problem of robust learning with label noises and focus on applying such methods to pre-trained language models like BERT.

Noise matrix is a transition matrix added to the end of DNNs to model the underlying label transition pattern of the noisy dataset (Sukhbaatar et al., 2014; Bekker and Goldberger, 2016; Patrini et al., 2017; Hendrycks et al., 2018; Yao et al., 2020; Jindal et al., 2019). Patrini et al. (2017) proposed Forward-Correction, which corrects wrong labels during forward propagation by multiplying the estimated noise transition matrix with the model's outputs. To obtain trustworthy noise matrices, Hendrycks et al. (2018) proposed gold loss correction that corrects loss using available trusted labels and then turns the confusion matrices of trusted labels into accurate transition matrices. Different from them, Jindal et al. (2019) trained transition matrices with an l_2 regularization which is not necessarily normalized into probability matrices. However, noise matrix methods have shown large estimation errors when only noisy data is available or when the noise level is high, which may not be feasible in real-world settings.

Regularization is widely used to prevent the overfitting of DNNs. Label smoothing (Szegedy et al., 2016) is such a method via softening ground truth labels by mixing the one-hot label with a uniform vector. As shown by Lukasik et al. (2020) and Zhang et al. (2017), label smoothing is an effective way to combat label noises. Jenni and Favaro (2018) proposed bilevel learning, which introduces a bilevel optimization using a clean validation dataset to regularize the overfitting of DNNs. However, the extended features introduced by regularization slow down the convergence of training, and the performance gain is very limited unless the models are deeper.

Robust Loss methods re-design the loss function to mitigate the negative impacts brought by incorrect labels. Kumar et al. (2020) has mathematically defined the pre-requisite for robust loss on multi-label classification tasks. Ghosh et al. (2017) showed the mean absolute error (MAE) loss satisfies such a condition and helps models achieve better generalization ability than the traditional cross-entropy loss. Wang et al. (2019) introduced symmetric cross entropy (SCE) loss that combines a reverse cross-entropy loss with the standard cross-entropy loss, achieving higher performance than previous methods. However, robust loss methods perform well only in simple cases where data patterns are easy to learn or the number of classes is small.

Co-Training is a family of methods that use two DNNs to help combat incorrect labels. Decoupling (Malach and Shalev-Shwartz, 2017) maintains two networks and updates them using instances with different predictions. Similarly, Co-teaching (Han et al., 2018) also trains two networks, but it selects small-loss data to teach the peer network, which is improved by Co-Teaching+ (Yu et al., 2019) through selecting small-loss data from only disagreement data. JoCoR (Wei et al., 2020) maintains two networks too, but it trains them together with a joint loss to maximize their agreement. However, the differences between two networks of the same architecture are very limited, especially during the later training period, so they can provide only slightly different views of the data. To solve such problems, Tan et al. (2021) proposed Co-Learning that introduced self-supervised learning to assist supervised learning of the classifier. However, the extra-introduced optimization largely slows down the convergence of the model.

Language models' robustness to label noises has not been as widely studied as Computer Vision models. Several attempts (Moradi and Samwald, 2021; Zhu et al., 2022a; Wang et al., 2020) have been made to improve language models' robustness to input perturbations, but they mainly focused on noisy data instead of noisy labels. Zhu et al. (2022b) showed that for text classification tasks with modern NLP models like BERT, existing noise-handling methods, including some methods mentioned above, do not always improve its performance under noisy labels of different noise rates, and may even deteriorate it. Jindal et al. (2019) proposed a CNN-based architecture that incorporates a non-linear processing layer to model the label noise statistics. But this method changes the commonly used NLP architecture, making pre-trained language models not usable, therefore it may be not applicable in various real-world corpora.

B Experiment Details

B.1 Baseline Descriptions

We compare SaFER with the following baselines:

- Without Noise-Handling, which does not apply any noise-handling modules to the classifier's training.
- Noise Matrix (Sukhbaatar et al., 2014), which appends a noise matrix after BERT's output to transform the clean label distribution to the noisy one.
- Noise Matrix with Regularization (Jindal et al., 2019), which also appends a noise matrix after BERT's output, but the matrix is trained with l2 regularization.
- Label Smoothing (Szegedy et al., 2016), which mixes each one-hot label with a uniform vector.
- Robust Loss, which leverages robust loss function (Mean Absolute Error Loss (Ghosh et al., 2017)) or designs new loss function (Symmetric Cross Entropy Loss (Wang et al., 2019)).
- Co-Teaching (Han et al., 2018), which trains two networks to select "clean" training subsets for each other.
- Co-Learning (Tan et al., 2021), which trains a projector along with the classifier to apply constraints on the classifier's learning.

The time per training step shown in Table 1 and Table 2 is calculated by averaging the total training duration across all label noise types on each dataset, including the time for model loading, training, and validation.

B.2 Hyperparameters

We set the following hyperparameters for SaFER evaluation:

Field	Value
BERT dropout rate	0.1
number of training steps (stage 1)	5000
number of training steps (stage 2)	5000
training batch size	32
evaluation batch size	64
evaluation frequency	25
feature dimension (projection)	128
number of batches for LID estimation	10
initial LID calculation step	5
LID window size	5
BERT learning rate	2e-5
SGD momentum	0.9
SGD dampening	0
SGD weight decay	0.0005
SGD nesterov	True
patience for early stopping	25

C Ablation Study Results

Here, we report the ablation study results in Section 3.3. We compare SaFER with one-stage pure BERT and one-stage BERT with unsupervised learning. The results is shown in Figure 3. We studied whether to apply early stopping to stage 2 to find the converging point of the classifier. The result is shown in Table 4. We also investigate the two early stopping strategies of fine-tuning the BERT classifier in noisy sets and evaluation in clean sets. The result is shown in Figure 4.

stage 1	stage 2	early stop stage 2?	accuracy (%)
BERT w/o noise-handling	BERT+unsup	yes	**89.06**
BERT w/o noise-handling	BERT w/o noise-handling	yes	84.49
BERT+unsup	BERT+unsup	yes	77.33
BERT w/o noise-handling	BERT+unsup	no	**90.01**
BERT w/o noise-handling	BERT w/o noise-handling	no	62.64
BERT+unsup	BERT+unsup	no	88.59

Table 4: Ablation study on the two-stage scheme.

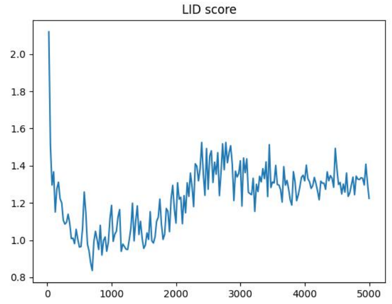


Figure 2: A typical LID curve in the label noise problem. Recorded at every 25 steps for training BERT-based classifier (without noise-handling) on IMDB dataset with low noise level ($\rho = 0.2$).

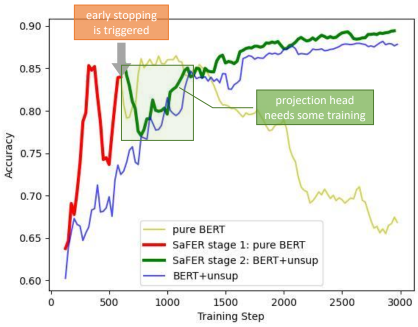


Figure 3: Comparing two-stage scheme with one-stage counterparts. Pure BERT method first gets to a high point and then drops significantly, but unsupervised learning could help avoid such a problem. Results are recorded when training on IMDB dataset with a high noise level ($\rho = 0.4$). The green box denotes the necessary steps for the projection head to catch up with the training.

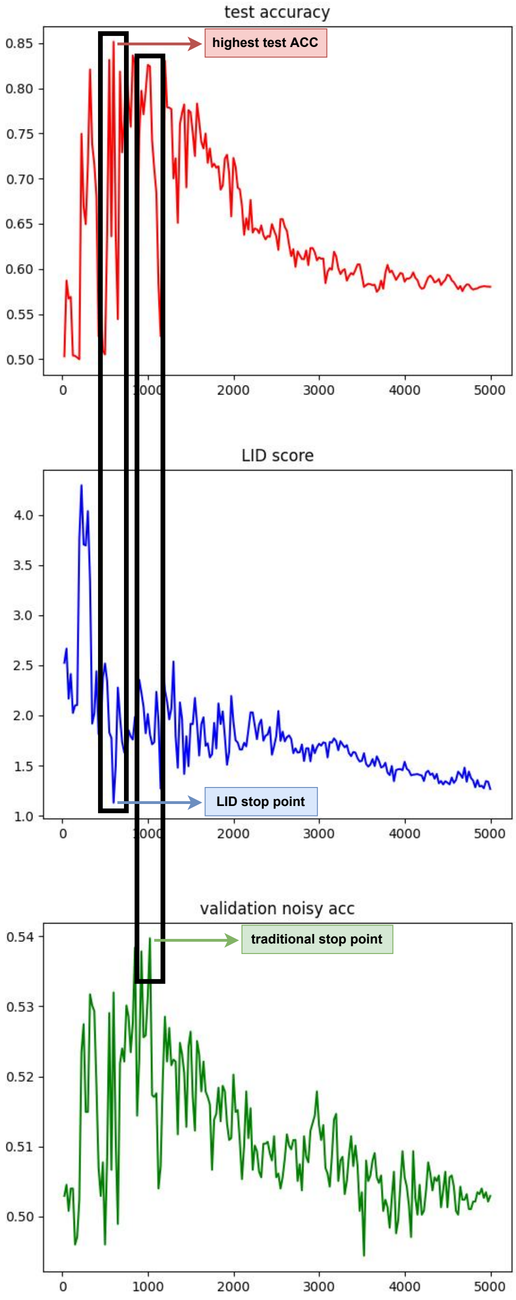


Figure 4: Comparing two early stopping strategies' results of fine-tuning BERT classifier without handling noise on IMDB (fast test), with extreme noise level (ρ =0.45), recorded for every 25 steps. The LID-based stopping strategy (middle) stops at the correct time, while the traditional strategy (bottom) misses the highest point.

D Algorithm

We demonstrate the full algorithm of SaFER in Algorithm 1.

Algorithm 1 SaFER: Two-Stage Finetuning

```
Input: Noisy training corpus $\widehat{C}$, pre-trained BERT backbone $f(\cdot;\theta_1)$, batch size $b$, number of stage
    training steps $T_1, T_2$ for stage 1 and 2
Output: Trained text classifier $f \cdot g$
  {STAGE 1: Fast Warming Up}
Initialize classification head $g(\cdot;\theta_2)$.
for each $t$ from 0 to $T_1 - 1$ do
    Sample a batch from $\widehat{C}$:
    $x \leftarrow \{x_i\}_{i=1}^{b}, \hat{y} \leftarrow \{\hat{y}_i\}_{i=1}^{b}$
    Obtain predictions:
    $u \leftarrow g(f(x;\theta_1);\theta_2)$
    Update $\theta_1, \theta_2$ using Eq. 3.
    Calculate batch LID scores using Eq. 2 and get the average score $lid_{avg}$.
    if $lid_{avg}$ reaches turning point then
        Save $\theta_1, \theta_2$ and break.
    end if
end for
  {STAGE 2: Combating Label Noises}
Initialize projection head $h(\cdot;\theta_3)$.
for each $t$ from 0 to $T_2 - 1$ do
    Sample a batch from $\widehat{C}$ and get a copy:
    $x \leftarrow \{x_i\}_{i=1}^{b}, \hat{y} \leftarrow \{\hat{y}_i\}_{i=1}^{b}; x' \leftarrow x$
    Encode sentences with BERT:
    $z \leftarrow f(x;\theta_1), z' \leftarrow f(x';\theta_1)$
    Get predictions from classifier $g$:
    $u \leftarrow g(z;\theta_2)$
    Get projections from projector $h$:
    $v \leftarrow h(z;\theta_3), v' \leftarrow h(z';\theta_3)$
    Update $\theta_2$ using Eq. 5.
    Update $\theta_1, \theta_2$ using Eq. 3.
    Update $\theta_1, \theta_3$ using Eq. 4.
end for
```