Model-agnostic Methods for Text Classification with Inherent Noise

Kshitij Tayal *
University of Minnesota
Twin Cities
tayal@umn.edu

Rahul Ghosh *
University of Minnesota
Twin Cities
ghosh128@umn.edu

Vipin Kumar
University of Minnesota
Twin Cities
kumar001@umn.edu

Abstract

Text classification is a fundamental problem, and recently, deep neural networks (DNN) have shown promising results in many natural language tasks. However, their human-level performance relies on high-quality annotations, which are time-consuming and expensive to collect. As we move towards large inexpensive datasets, the inherent label noise degrades the generalization of DNN. While most machine learning literature focuses on building complex networks to handle noise, in this work, we evaluate model-agnostic methods to handle inherent noise in large scale text classification that can be easily incorporated into existing machine learning workflows with minimal interruption. Specifically, we conduct a point-by-point comparative study between several noise-robust methods on three datasets encompassing three popular classification models. To our knowledge, this is the first time such a comprehensive study in text classification encircling popular models and model-agnostic loss methods has been conducted. In this study, we describe our learning and demonstrate the application of our approach, which outperformed baselines by up to 10 % in classification accuracy while requiring no network modifications. Code for this paper is hosted at www.kshitijtayal.com/code/model-agnostic-methods.

1 Introduction

Text classification is a fundamental problem in natural language processing, where the objective is to categorize text into a set of predefined classes. It has been shown to be valuable in many domains, such as social media (Kateb and Kalita, 2015), cognitive-biometric recognition (Pokhriyal et al., 2016) and e-commerce (Yu et al., 2012). Modern-day enterprises are heavily dependent on the performance of text classification models, where even a marginal improvement in the performance can accrue billions of dollars (Singh, 2019) and substantially improve the customer experience.

Currently, DNN (Zhou et al., 2016; Devlin et al., 2018) are the state of the art machine learning models widely deployed for text classification tasks in major enterprises (Bernardi et al., 2019; Haldar et al., 2019; Liu et al., 2019). Like any other supervised classifiers, the performance of these DNN trained using standard cross-entropy loss is strongly dependent on the quality and quantity of the data. However, collecting high-quality manual labels is time-consuming and expensive. At the same time, there are less expensive sources to collect labeled data, such as Mechanical Turk (Kittur et al., 2008), search engine meta data, and social media tags. These inexpensive large datasets have a high level of noise, as multiple annotators generate the labels under different skill-set and biases. In e-commerce, an example of one such confusing case is when the same product title is labeled differently by agents into separate but related categories, as shown in Fig. 1. Blindly trusting these large inexpensive datasets as gold-standard can decrease the performance of models.

Learning from noisy labels is an active area of research in computer vision, and several model cognizant approaches (Wu et al., 2018; Lefkimmiatis, 2018) have been proposed. However, these approaches work on building complex network architecture to handle noise and require substantial back-

*equal contribution

Proceedings of the 28th International Conference on Computational Linguistics: Industry Track, pages 202–213
Barcelona, Spain (Online), December 12, 2020

ground knowledge and training to operate. For many enterprises, the performance of text classification models plays a crucial role in their revenue earnings, and the difficulty of implementing complex architecture becomes a bottleneck. Conversely, there is minimal research studying the performance of the model-agnostic methods to handle inherent label noise, that can be easily incorporated into existing machine learning workflows with no network modifications for large scale text classification tasks.

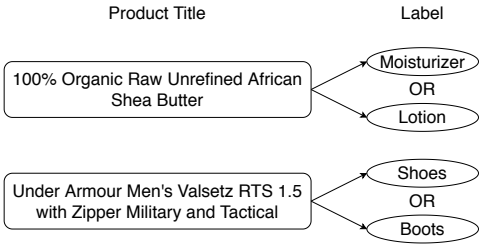

Figure 1: Noisy labels arising due to labels assigned by multiple annotators.

Under model agnostic schemes, there are several different lines of work which include modeling noise-transition matrix (Patrini et al., 2017; Goldberger and Ben-Reuven, 2016; Sukhbaatar et al., 2014), training auxiliary network (Jiang et al., 2017; Guo et al., 2018), training with clean labels (Malach and Shalev-Shwartz, 2017), label regularization (Szegedy et al., 2016), data augmentation (Zhang et al., 2017), and noise-robust loss functions. In this work, we focus our attention on techniques that do not add any overhead computation. Specifically, we evaluate label smoothing regularization, data augmentation technique, and state of the art noise-robust loss functions (Reed et al., 2014; Wang et al., 2019; Ma et al., 2018; Zhang and Sabuncu, 2018) to examine its effect in mitigating inherent label noise for large scale text datasets. These methods are simpler and easy to implement than other lines-of-work in tackling noisy labels, which either gets very complex (Jiang et al., 2017) or has strong assumptions on the type of noise present (Sukhbaatar et al., 2014). We conduct our study on large web-scale text data scraped from popular e-commerce platforms, which contains a significant number of classes leading to a higher inherent noise due to annotator confusion. In contrast with previous work (Li et al., 2019; Jindal et al., 2019), we do not introduce any external noise into our dataset. To the best of our knowledge, no previous study has been done to study model-agnostic methods in mitigating inherent label noise for large scale text classification. To summarize, the main contributions of our work are as follows:

- We propose the use of model-agnostic methods to handle inherent noise in the context of text classification on large scale datasets. To our knowledge, this is the first attempt to use model-agnostic methods for text classification.
- We perform extensive experiments on three real-world datasets scraped from popular e-commerce platform. We show that our approach outperforms baselines with a margin of 10% in classification accuracy using three popular classification models.

2 Related Work

In this section, we provide a brief literature review for model agnostic methods popularly used in machine learning to handle noise. These include modeling label noise, training auxiliary network, data augmentation, noise-robust loss functions, and regularization schemes.

Existing literature in modeling label noise can be further subdivided into two groups: class-conditional and instance-conditional label noise. The first group assumes that the noise is independent of the instance and models the transition probability from true class to noisy class. (Mnih and Hinton, 2012) assumed the class-conditional label noise for binary classification task and consequently use an EM-based algorithm to learn the model parameters and the noise transition matrix. (Sukhbaatar et al., 2014) extended the multi-class counterpart of class-conditional noise and proposed a constrained linear layer at the top of the softmax layer, which under some strong assumptions can be interpreted as the noise transition matrix. A similar work by (Patrini et al., 2017) uses forward and backward methods to explicitly model the noise transition matrix and also provide a way to estimate the noise transition matrix. The second group assumes that the label noise is conditioned for each instance. (Xiao et al., 2015) developed a noise model, where noise is modeled on the instance and its class. Similarly, (Vahdat, 2017) model the noise through Conditional Random Fields (CRF), where the clean labels are modeled as latent variables during training.

Under training auxiliary network, (Malach and Shalev-Shwartz, 2017) proposed training two different networks which back-propagate the loss when the predictions of the two network disagree. Mentor network (Jiang et al., 2017), another popular method, learns a sample weighting scheme to supervise the training of a base network, termed StudentNet, that learns under label noise contingencies. Similarly, (Guo et al., 2018) present an unsupervised approach to learn the curriculum based on the complexity of the instance in the feature space. Supporting the loss function category, (Natarajan et al., 2013) presented robust surrogate loss functions for handling noisy labels in a binary classification task. Mean absolute error (MAE) (Ghosh et al., 2017) was shown to be inherently robust to label noise for the classification task. Similarly bootstrapping loss function (Reed et al., 2014) was proposed, which introduced a weighted combination of target labels and network predictions to compensate for noisy samples. While (Reed et al., 2014) uses a fixed hyper-parameter as weights, D2L (Ma et al., 2018) proposes to use the subspace complexity score of the model as weights which gets updated at every iteration. To overcome the limitation of MAE, Generalized Cross Entropy (Zhang and Sabuncu, 2018) was proposed, which is a combination of MAE and categorical cross entropy (CCE) loss. Symmetric cross-entropy (Wang et al., 2019) augments the standard CCE, similar to symmetric KL-divergence, with the noise robust reverse cross-entropy.

Data augmentation and regularization schemes are some other ways introduced to make the learning procedure robust to noisy labels. These include, mixup (Zhang et al., 2017), that uses convex combinations of training data points and its corresponding labels, and Label Smoothing Regularization (LSR) (Szegedy et al., 2016), where a smoothing parameter is used to modify the hard one-hot labels into soft labels to mitigate over-fitting to noisy labels.

3 Method

3.1 Problem Setting

In this paper we consider the text classification problem where each data instance is described as features $\boldsymbol{x} \in \mathbb{R}^d$ and label $\boldsymbol{y} \in \{0,1\}^K$ (one hot encoded vector). $\boldsymbol{x}$ is the vector representation of a text, where d is the dimensionality of the embedding vector and K are the number of classes.

Table 1: Summary statistics of datasets

Dataset	#Samples	#Unique Words	#Class
Beauty	207574	118215	342
Electronics	362142	424368	823
Automotive	243296	228234	1818

Data: We conduct our study on large web-scale product title data scraped from Amazon (He and McAuley, 2016)[1]. The datasets are broken down by categories, and we make use of three such categories i.e. **Electronics**, **Beauty** and **Automotive**. Table 1 shows the characteristics of these datasets. Each dataset contains product titles, metadata for each product (also bought, also viewed, bought together, buy after viewing), and their categories. For each product, its category is a path from a coarse-grained label to a fine-grained label. We use the product titles as inputs and the fine grained label from the above metadata as the product label. E.g., a product in category `Electronics` ⇒ `Computers & Accessories` ⇒ `Cables & Accessories`, will have `Cables & Accessories` as its label. We do a 70:30 split of our dataset $\mathcal{D} = \{\boldsymbol{x}_i, \boldsymbol{y}_i\}_{i=1}^N$ into training set $\mathcal{D}^{Train}$ and test set $\mathcal{D}^{Test}$ such that $\{\mathcal{D}^{Train} \cap \mathcal{D}^{Test} = \emptyset\}$ and $\{\mathcal{D}^{Train} \cup \mathcal{D}^{Test} = \mathcal{D}\}$.

Goal: Our objective is to learn a classification model $f(\boldsymbol{x}, \theta)$ on the training set $\mathcal{D}^{Train}$ which learns an accurate mapping function f such that it makes correct prediction on test sample $\boldsymbol{x}_i \in \mathcal{D}^{Test}$. Here θ are the parameters of the DNN.

3.2 Model-agnostic Methods

The underlying principle of training classification models is to minimize a loss function and accordingly update the network parameters. In the classification task, categorical cross entropy (CCE) loss is one such loss function which measures the performance of a classification model whose output is a likelihood

[1] http://jmcauley.ucsd.edu/data/amazon/links.html

estimation $f(\boldsymbol{x};\theta)$ between 0 to 1 scale. The CCE loss is given by

$$\mathcal{L}_{CCE} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{K} y_{ij} log(f_j(\boldsymbol{x}_i;\theta)) \tag{3.1}$$

where, y_{ij} is the j'th element of $\boldsymbol{y}_i$. The features, label and network prediction of the i'th instance are denoted by $\boldsymbol{x}_i$, $\boldsymbol{y}_i$ and $f(\boldsymbol{x}_i;\theta)$ respectively. K is the number of classes and N is the number of training examples. The number of parameters in most deep architectures are very large and often exceeds the size of the data used for training. There is significant theoretical and empirical evidence that in such over-parametrized DNN, the output of the trained model matches the training labels exactly (Zhang et al., 2016). Consequently, if the training labels contain noise, the learned weights can be sub-optimal leading to high test error in-spite of low training losses. In the following segment, we briefly describe noise-robust learning methods we used in our evaluation to overcome inherent noise in large datasets.

3.2.1 Label Smoothing Regularization (LSR)

CCE loss encourages the model to be more confident on its predictions by minimizing the probabilities of the given class which can be particularly harmful in case of noisy labels, as the model overfits on the noisy examples resulting in poor generalization performance. To regularize the model and make it more adaptable, (Szegedy et al., 2016) proposed to use a mixture of the original ground truth distribution with another fixed distribution u in place of the original labels. The target label is modified as $y'_{ij} = (1-\epsilon)y_{ij} + \epsilon u(j)$, where, $u(j)$ is used as a fixed prior distribution over labels weighted by ϵ. Thus, using this weighted target label, the loss function takes the following form

$$\mathcal{L}_{LSR} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{K} [(1-\epsilon)y_{ij} log(f_j(\boldsymbol{x}_i;\theta)) + \epsilon u(j) log(f_j(\boldsymbol{x}_i;\theta))] \tag{3.2}$$

3.2.2 Bootstrapping

Proposed by (Reed et al., 2014), Bootstrapping loss function expands the prediction objective with a notion of consistency. A prediction is consistent if an identical prediction is made given similar percepts, where the idea of similarity is between model features estimated from the input data. Bootstrapping loss function dynamically updates the target labels by using a convex combination of the current model's prediction and the (possibly noisy) training label. The weight of the convex combination is administered by hyperparameter β. This process provides the model justification to "disagree" with inconsistent training label, and efficiently re-label the data while training. This approach is referred to as soft bootstrapping when the predicted probabilities are directly used to generate target labels as follows

$$\mathcal{L}_{boot\text{-}soft} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{K} [\beta y_{ij} + (1-\beta) f_j(\boldsymbol{x}_i;\theta)] log(f_j(\boldsymbol{x}_i;\theta)) \tag{3.3}$$

Similarly, the approach is referred to as hard bootstrapping when the predicted class probabilities are replaced by their one-hot encoded vector based on the maximum apriori probability (MAP) estimate as follows

$$\mathcal{L}_{boot\text{-}hard} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{K} [\beta y_{ij} + (1-\beta) z_{ij}] log(z_{ij}) \tag{3.4}$$

where, $\boldsymbol{z}_i = \mathbb{1}[k = \text{argmax } f_j(\boldsymbol{x}_i;\theta), j = i \ldots K]$

3.2.3 Mixup

(Zhang et al., 2017) proposed a simple data augmentation technique that works on the vicinal risk minimization principle (Chapelle et al., 2001), where virtual data instances created in the vicinity of training data instances are used for risk minimization. Mixup constructs virtual training examples under the

assumption that linear interpolation of feature vectors should lead to linear interpolation of associated targets and thus takes the form

$$\tilde{x} = \lambda \boldsymbol{x}_i + (1-\lambda)\boldsymbol{x}_j \qquad \tilde{y} = \lambda \boldsymbol{y}_i + (1-\lambda)\boldsymbol{y}_j \tag{3.5}$$

where, $\boldsymbol{x}_i, \boldsymbol{x}_j$ are raw feature vectors and $\boldsymbol{y}_i, \boldsymbol{y}_j$ are the corresponding one-hot labels. λ is sampled from a beta distribution Beta(α, α), for $\alpha \in (0, \infty)$. Increasing α results in virtual examples further from the training examples. The authors hypothesize that learning linear interpolations of real instances is easier than memorizing random noisy labels and thus this strategy should avoid the model to overfit to the corrupted labels.

3.2.4 Dimensionality Driven Learning (D2L)

(Ma et al., 2018) introduced a new prospect for understanding DNN generalization by examining the dimensionality of the representation subspace of training samples. They explain that DNN exhibits a two-stage learning style when training with noisy labels, i.e., 1) an early stage of dimensionality compression that models low dimensional subspace that approximately resembles the underlying distribution and 2) a later stage of dimensionality expansion that expands subspace dimensionality to overfit noisy labels. Thus, to avoid noisy labels, a label smoothing strategy is proposed, which finds an optimal trade-off between the model prediction and the training labels. Specifically, the model is trained with the training labels until a turning point is found, at which point the model starts to overfit. This turning point is determined based on Local Intrinsic Dimensionality (LID) (Houle, 2017), which is a measure of the subspace dimensionality at each epoch. Specifically, at any epoch for a training instance $\boldsymbol{x}$, LID is calculated as:

$$LID(\boldsymbol{x}, X_B) = -\left(\frac{1}{k} \sum_{i=1}^{k} \log \frac{r_i(g(\boldsymbol{x}), g(X_B))}{r_{max}(g(\boldsymbol{x}), g(X_B))} \right)^{-1} \tag{3.6}$$

where, X_B is a random batch selected from the training set, g is the second-to-last DNN layer, $r_i(g(\boldsymbol{x}), g(X_B))$ is the distance between $\boldsymbol{x}$ and its i-th nearest neighbor in the transformed space and r_{max} is the largest value among the k nearest neighbors thus denoting the radius of the neighborhood. After the turning point is established, the training labels are smoothed by adding the network prediction to them, and these smoothed labels are used for training the models. Smoothed labels are calculated as follow:

$$\overset{*}{y} = \alpha_t y + (1-\alpha_t)\hat{y}, \text{ where } \quad \alpha_t = exp\left(-\lambda \frac{LID_t}{\min_{j=0}^{t-1} LID_j} \right) \tag{3.7}$$

is a LID-based factor that updates at the t-th training epoch. y is the raw label, $\hat{y}$ is the predicted label, and $\lambda = j/T$, (T: total epochs) is a weighting that indicates diminishing confidence in the raw labels when the training proceeds to the dimensionality expansion stage. Dimensional expansion is evaluated in terms of the ratio of two average LID scores: the current epoch's score, and the lowest score encountered at earlier epochs. The ratio exceeds one as the learning enters the dimensional expansion stage, and after that, the exponential decay factor starts to support the current model prediction. The training loss can then be refined as:

$$\mathcal{L}_{D2L} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{K} \overset{*}{y}_{ij} log(f_j(\boldsymbol{x}_i; \theta)) \tag{3.8}$$

where, N is the total number of training samples.

3.2.5 Generalized Cross Entropy (GCE)

Proposed by (Zhang and Sabuncu, 2018), GCE is a generalization of CCE and mean absolute error (MAE) with hyperparameter q, where $q \in [0, 1]$. When $q \to 0$, the loss becomes CCE, and likewise becomes MAE/unhinged loss when $q = 1$. During training with CCE, the loss function implicitly

puts more stress on samples where the model disagrees with the target labels, which is useful when training data is clean but can cause overfitting to noisy labels. Conversely, MAE weighs all predictions equally, which makes it more robust to noisy labels (Ghosh et al., 2017). However, in our experiments with product title classification tasks, we see that the neural network was not able to converge and gave an abysmal result on the test dataset. This finding is coherent with other authors' works (Fonseca et al., 2019; Zhang and Sabuncu, 2018). GCE addressed the challenge by taking advantage of both the noise-robustness provided by MAE and the implicit weighting scheme of CCE. The GCE loss is given by

$$\mathcal{L}_{GCE} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{K}\frac{1-(y_{ij}f_j(\boldsymbol{x}_i;\theta))^q}{q}, q \in [0,1] \tag{3.9}$$

3.2.6 Symmetric Cross Entropy (SL)

Cross-entropy by itself is not sufficient for learning generalizable models in presence of noisy labels. The training labels don't represent the true class, whereas after a few iterations of training the model output can start to get closer to the true class distribution. Therefore, in addition to the standard CCE, (Wang et al., 2019) propose to use the reverse cross entropy (RCE) in the loss function and the final loss is a weighted combination of both as given below

$$\mathcal{L}_{SL} = \alpha\mathcal{L}_{CCE} + \beta\mathcal{L}_{RCE} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{K}\alpha y_{ij}log(f_j(\boldsymbol{x}_i;\theta)) + \beta f_j(\boldsymbol{x}_i;\theta)log(y_{ij}) \tag{3.10}$$

where, α and β are two hyperparameters. Here, the CCE loss helps in convergence whereas the RCE loss is noise tolerant and penalizes the model predictions that has been optimized for the noisy training labels.

4 Experiments and Results

In this section, we evaluate several model-agnostic approaches discussed above on three large scale datasets shown in Table 1 and attempt to answer the following questions:

- Do model-agnostic methods give a substantial gain in performance for large web-scale data having inherent noise over baseline?
- How does the performance vary for model-agnostic methods under different types of models?
- How the behavior of model-agnostic methods change as we introduce external noise? Is there any correlation between performance gain and the number of class label?

4.1 Learning Models

In this section, we provide a brief discussion of the models used in our comparative study. **FFNN:** In this work, we use FFNN (Rumelhart et al., 1985) with average pooling operation (Shen et al., 2018) on input feature with two hidden layers having 1024 and 512 units respectively. We employ a ReLU activation function for non linearity with 0.2 dropout followed by a output layer of K output values, where K is the number of classes. **CNN:** 1D CNN (Kim, 2014) and fixed the maximum length of sentence to 10 and embedding dimension 128. In our network architecture, we use one convolutional layer having 128 filters with a convolution window/kernel size of 5 followed by max-pooling and finally a fully connected layer with 512 neurons. **LSTM:** For LSTM (Hochreiter and Schmidhuber, 1997), we use the same sentence length and embedding size as CNN. In our network architecture, the first layer of LSTM is the embedding layer

Table 2: Hyperparameters for model-agnostic methods

Method	Hyper-parameters
LSR	$\epsilon = 0.3$
$Boot\text{-}hard$	$\beta = 0.3$
$Boot\text{-}soft$	$\beta = 0.3$
$mixup$	$\alpha = 0.2$
GCE	$q = 0.3$
SL	$\alpha = 2, \beta = 1$

Table 3: Relative performance of different model-agnostic methods against cross-entropy loss with no external noise

MODEL	DATASET	$CCE(\%)$	LSR	$BootHard$	$BootSoft$	$mixup$	$D2L$	GCE	SL
FFNN	Beauty	68.16	0.79	0.1	1.19	**2.63**	2.45	2.43	1.13
	Electronics	70.36	0.91	0.2	1.09	2.57	**3.49**	3.18	1.62
	Automotive	73.19	1.82	0.42	1.08	**3.02**	1.83	1.42	1.31
LSTM	Beauty	68.76	1.31	1.05	1.69	1.59	1.81	**1.98**	1.28
	Electronics	66.16	2.03	0.53	1.59	**2.86**	2.55	1.63	1.81
	Automotive	73.50	1.24	0.34	0.79	2.34	**3.73**	1.09	1.32
CNN	Beauty	60.33	3.93	4.19	3.93	3.91	**6.41**	5.37	3.08
	Electronics	56.79	5.35	1.95	4.75	4.67	**9.8**	5.6	4.37
	Automotive	64.36	4.3	2.41	2.95	3.59	**10.74**	9.4	3.56

followed by variational dropout. The next layer is the LSTM layer, with 256 memory units, followed by the output layer of K output values. More recently, researchers have started to apply graph convolutional networks (Tayal et al., 2019; Yao et al., 2019) for text classification. Preliminary results are encouraging; however, they bring in the additional complexity of the graph. In this work, we restrict ourselves to more popular techniques, i.e., FFNN, CNN, LSTM.

4.2 Experimental setup

We trained all models for a maximum of 75 epochs using Adam optimizer (Kingma and Ba, 2014) with 0.001 learning rate and terminate training if the validation loss does not reduce for 10 continuous epochs. To remove bias between different model runs, the train, validation, and test set are kept consistent for all models. We refer the model trained on cross-entropy loss as baseline. Individual words are encoded using glove embeddings (Pennington et al., 2014). For electronics dataset, we fixed hyperparameter for each of the methods using grid search based on their average performance on 5 fold cross-validation. Due to constraints in the use of computational hardware, we fixed the same hyperparameters (Table 2) for other datasets too.

4.3 Results

Table 3 reports the relative performance of the different model-agnostic methods. The column for CCE shows the absolute baseline accuracy, and other columns represent the percentage improvement achieved by model-agnostic methods over their cross-entropy trained counterpart. We highlight best performing methods for each row and make the following high-level observations from our results: a) All values in result table 3 are positive, which strengthens our statement that there is inherent noise in large text datasets, which can result in overfitting of DNNs trained on standard CCE loss, and b) $D2L$ is the top-performing method that gave the best result consistently over CCE, followed by GCE and $mixup$. Likewise, *boot-hard* and *boot-soft* worked well over CCE but not as high as other methods.

Continuing to expand on the above observations, $D2L$ is the best performing model, which suggests that dimensionality driven learning strategy is highly tolerant to noisy labels and works best for large scale text classification. The performance improvement is much more visible when CNN is used in conjunction with $D2L$.

GCE, which is a generalization of cross-entropy and MAE has comparable performance with $D2L$ and consistently outperforms CCE. This shows the benefit of using the noise-robustness feature of MAE in conjunction with CCE.

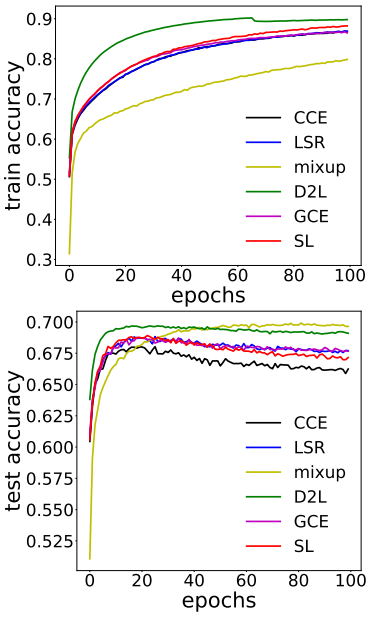

Figure 2: Train and Test accuracy against number of epochs for FFNN on Beauty dataset

Table 4: Relative performance of different model-agnostic methods against cross-entropy loss with 20% noise

MODEL	DATASET	$CCE(\%)$	LSR	$Boot\text{-}hard$	$Boot\text{-}soft$	$mixup$	$D2L$	GCE	SL
FFNN	Beauty	66.42	0.53	1.2	1.84	2.12	**2.78**	2.78	1.25
	Electronics	68.80	0.83	0.58	1.09	2.81	**3.62**	3.38	1.9
	Automotive	70.90	1.76	1.71	2.75	3.03	**4.49**	2.91	2.74
LSTM	Beauty	67.11	1.49	**3.01**	1.22	0.86	1.80	2.94	1.71
	Electronics	63.73	2.73	-1.6	0.41	3.69	**5.60**	4.06	3.0
	Automotive	70.90	2.0	0.71	2.28	2.38	**4.59**	3.07	2.02
CNN	Beauty	54.71	7.27	10.38	11.99	5.1	**13.03**	12.5	6.62
	Electronics	53.33	5.57	-4.56	1.5	9.52	**15.15**	9.1	5.18
	Automotive	59.06	6.3	4.25	6.86	10.04	**16.44**	10.68	5.81

Likewise, SL uses a combination of reverse cross-entropy, which adds value when used in conjunction with CCE. $mixup$, a simple data augmentation technique gave impressive gain for FFNN and LSTM. However, it didn't perform well on CNN as compared to other methods, which showcase the gap in learning when hundreds of unique class labels are present. Likewise, LSR has average performance gains due to the huge number of classes present in our dataset. The huge number of classes reduces the label smoothing effect of the approach, which relies on the addition of a fixed uniform label distribution to the one-hot labels.

$Boot\text{-}hard$ and $Boot\text{-}soft$ performed fine, but not as high as other methods. We attribute this to hyperparameter β, which is fixed for each epoch and controls the convex combination of the model prediction and the training label. $D2L$, on the other hand, overcame this and set its parameter for each epoch in an automated fashion using model complexity.

Although our goal is not to compare the performance between different models, we cannot help but notice that for the Automotive dataset, $D2L$ and GCE were able to bring the performance of CNN closer to that of FFNN. We thereby conclude that in some cases model-agnostic methods can further help to make existing models more powerful.

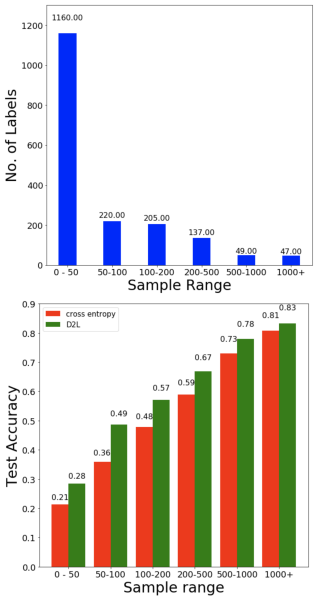

Figure 3: Top: distribution of labels with respect to the sample range for the automotive dataset. Bottom: Test accuracy comparison b/w CCE loss and D2L across different sample range.

4.4 Accuracy Curves

Figure 2 denotes training and test accuracies at every epoch attained by FFNN on the beauty dataset. We observe that the classifier trained using CCE first learns discriminative patterns, which is evident from high test accuracy in the initial epochs. Later the test accuracy decreases as the model starts overfitting on the noisy labels, which explains the increase in train accuracy (CCE training curve overlapped by LSR). This validates report from other works (Zhang et al., 2016; Arpit et al., 2017) that DNNs first learn predictive patterns from easily separable instances and later overfits to the noisy labels. On the contrary, training with model-agnostic methods limits overfitting to noisy labels and achieved higher test accuracies.

Specifically, $D2L$ and $mixup$ are the most effective methods in limiting the overfitting effect. We note that low training accuracy of $mixup$ is on linear interpolated data, while test accuracy is on original test samples. These observations serve as an empirical justification for the use of model-agnostic approaches.

4.5 Noise Robustness

To further evaluate the performance of model-agnostic methods, we randomly flipped 20% training labels and compared the model performance on the test set where the labels are not touched. Table 4 reports the performance when external noise is added. All the settings are the same as in Section 4.3. As with no noise, we note that $D2L$ is consistently the top performer, and the margin becomes more distinct. Specifically, if we look at CNN, it gave 13.03%, 15.15%, and 16.44% gain for beauty, electronics, and automotive dataset, respectively, which translates to absolute performance of 61.82 %, 61.32 %, 68.51%. These numbers are close to the result given by $D2L$ when no noise was present, concluding that $D2L$ is more stable in the presence of noise. We also observe that $Boot\text{-}hard$ loss function breaks down when we increase the noise, particularly when used with LSTM and CNN in the Electronics dataset.

We further experimented by flipping 40% training labels and find that most of the approaches gave an inferior performance as compared to CCE, thereby concluding that model-agnostic methods do not perform well with very large noise.

4.6 Data Imbalance

As with any large dataset with hundreds of classes, our data set is imbalanced (Figure 3 -top), purporting that we have more classes with fewer samples and fewer classes with more samples. In this section, we study whether the performance gain is uniform throughout the class space, or it changes with the number of samples/class. To evaluate, we select automotive dataset having 1818 labels and compare the performance of the CNN model trained using $D2L$ and CCE loss. Notably, we divide the classes into 6 categories according to the number of samples in the dataset i.e., 0-50, 50-100, 100-200, 200-500, 500-1000, 1000+. Figure 3 (bottom) displays the test accuracy for both the models across six categories. We observe that the highest gain in performance is achieved for the classes with 50-100 samples, followed by 100-200 samples, 200-500 samples. 1000+ samples make the lowest gain in performance. Thus from these observations, we reason that the model agnostic methods is more advantageous for classes having samples in the range 50-500, while classes having more samples gets a limited advantage.

4.7 Impact of number of class label

In this section, we investigate the relationship between class label size and performance gain. As the number of classes increases, it presents an additional complexity on model learning to learn the accurate boundary. We observe that some model-agnostic methods performance gain have positive correlation with the number of class label. Specifically, in Table 3, when $D2L$ is used with CNN, we observe performance gain of 6.41 %, 9.8 %, 10.74 %, which directly relates to class label size of 342, 823 and 1818 for beauty, electronics, and automotive dataset respectively. The same trend continues in table 4 when we flip 20% of the labels. We observe this trend owing to the fact that as the number of class labels increases, the annotator becomes more confused in labeling, which results in more inherent noise in the datasets. Thus the use of model-agnostic methods becomes more necessary when training machine learning models on large datasets with hundreds and thousands of categories.

4.8 Comparison with large expressive models

In this section, we compare model-agnostic method performance with a large expressive model like pre-trained BERT (Devlin et al., 2018). We consider three pre-trained BERT models (trained on Wikipedia and the Book Corpus dataset) having 2, 4, and 6 layers respectively and fine-tune them on our inputs using standard CCE loss. The models were fine-tuned for 600 epochs using a batch size of 500 and a learning rate of $1e-5$. Table 5 provides test error for all BERT model. From the results, we conclude that FFNN with $D2L$ can easily outperform the fine-tune BERT models (2-4-6 layers) consistently over all the three datasets, which reiterates the use of noise-robust loss functions. We can additionally increase the complexity of BERT models by adding more layers, but then the model will be highly com-

Table 5: BERT model performance

DATASET	L-2	L-4	L-6
Beauty	64.7	67.3	70.6
Electronics	56.2	62.7	65.2
Automotive	57.9	65.8	67.3

plex and cannot be used for inference in production. Due to computational hardware constraints, we leave for future work to explore such a model's performance and how these model-agnostic approaches will work for BERT when trained from scratch.

5 Conclusion

In this study, we demonstrate the effectiveness of model-agnostic methods in advancing the performance of machine learning models for large scale text classifications. While most of the machine learning literature focused on building complex networks to handle noise, very few works have studied the performance of simpler methods that can give a significant impact. To the best of our knowledge, this is the first attempt to apply model agnostic methods requiring no network modifications to handle inherent noise for text classification datasets. We fill the gap in existing literature, where applying these methods to large scale text classification tasks is not the norm. Although we have shown improvements for data scraped from e-commerce platforms, the methods mentioned above can be applied to any large text classification task. The methods mentioned are easy to implement and can be easily integrated into any machine learning workflows without breaking the existing code-base. In contrast to previous works, we did not add noise and hypothesize that large dataset having thousands of samples and hundreds of unique classes can inadvertently introduce noise. Moreover, this paper serves as a brief literature review of model-agnostic methods that can be applied to text classification and other related domains, requiring no network modifications and minimal computation overhead.

Acknowledgement

This research was supported by National Science Foundation under the grant 1838159 and 1739191. Access to computing facilities was provided by the University of Minnesota Supercomputing Institute.

References

Devansh Arpit, Stanislaw Jastrzkbski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, et al. 2017. A closer look at memorization in deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 233–242. JMLR. org.

Lucas Bernardi, Themistoklis Mavridis, and Pablo Estevez. 2019. 150 successful machine learning models: 6 lessons learned at booking. com. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1743–1751.

Olivier Chapelle, Jason Weston, Léon Bottou, and Vladimir Vapnik. 2001. Vicinal risk minimization. In *Advances in neural information processing systems*, pages 416–422.

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*.

Eduardo Fonseca, Manoj Plakal, Daniel PW Ellis, Frederic Font, Xavier Favory, and Xavier Serra. 2019. Learning sound event classifiers from web audio with noisy labels. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 21–25. IEEE.

Aritra Ghosh, Himanshu Kumar, and PS Sastry. 2017. Robust loss functions under label noise for deep neural networks. In *Thirty-First AAAI Conference on Artificial Intelligence*.

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

Sheng Guo, Weilin Huang, Haozhi Zhang, Chenfan Zhuang, Dengke Dong, Matthew R Scott, and Dinglong Huang. 2018. Curriculumnet: Weakly supervised learning from large-scale web images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 135–150.

Malay Haldar, Mustafa Abdool, Prashant Ramanathan, Tao Xu, Shulin Yang, Huizhong Duan, Qing Zhang, Nick Barrow-Williams, Bradley C Turnbull, Brendan M Collins, et al. 2019. Applying deep learning to airbnb search. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1927–1935.

Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *proceedings of the 25th international conference on world wide web*, pages 507–517. International World Wide Web Conferences Steering Committee.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Michael E Houle. 2017. Local intrinsic dimensionality ii: multivariate analysis and distributional support. In *International Conference on Similarity Search and Applications*, pages 80–95. Springer.

Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. 2017. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. *arXiv preprint arXiv:1712.05055*.

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

Faris Kateb and Jugal Kalita. 2015. Classifying short text in social media: Twitter as case study. *International Journal of Computer Applications*, 111(9).

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Aniket Kittur, Ed H Chi, and Bongwon Suh. 2008. Crowdsourcing user studies with mechanical turk. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 453–456.

Stamatios Lefkimmiatis. 2018. Universal denoising networks: a novel cnn architecture for image denoising. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3204–3213.

Junnan Li, Yongkang Wong, Qi Zhao, and Mohan S Kankanhalli. 2019. Learning to learn from noisy labeled data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5051–5059.

Bang Liu, Weidong Guo, Di Niu, Chaoyue Wang, Shunnan Xu, Jinghong Lin, Kunfeng Lai, and Yu Xu. 2019. A user-centered concept mining system for query and document understanding at tencent. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1831–1841.

Xingjun Ma, Yisen Wang, Michael E Houle, Shuo Zhou, Sarah M Erfani, Shu-Tao Xia, Sudanthi Wijewickrema, and James Bailey. 2018. Dimensionality-driven learning with noisy labels. *arXiv preprint arXiv:1806.02612*.

Eran Malach and Shai Shalev-Shwartz. 2017. Decoupling" when to update" from" how to update". In *Advances in Neural Information Processing Systems*, pages 960–970.

Volodymyr Mnih and Geoffrey E Hinton. 2012. Learning to label aerial images from noisy data. In *Proceedings of the 29th International conference on machine learning (ICML-12)*, pages 567–574.

Nagarajan Natarajan, Inderjit S Dhillon, Pradeep K Ravikumar, and Ambuj Tewari. 2013. Learning with noisy labels. In *Advances in neural information processing systems*, pages 1196–1204.

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.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Neeti Pokhriyal, Kshitij Tayal, Ifeoma Nwogu, and Venu Govindaraju. 2016. Cognitive-biometric recognition from language usage: A feasibility study. *IEEE Transactions on Information Forensics and Security*, 12(1):134–143.

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*.

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. 1985. Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science.

Dinghan Shen, Guoyin Wang, Wenlin Wang, Martin Renqiang Min, Qinliang Su, Yizhe Zhang, Chunyuan Li, Ricardo Henao, and Lawrence Carin. 2018. Baseline needs more love: On simple word-embedding-based models and associated pooling mechanisms. *arXiv preprint arXiv:1805.09843*.

Shelly Singh. 2019. Natural language processing market worth $26.4 billion by 2024. *https://www.bloomberg.com/press-releases/2019-12-10/natural-language-processing-market-worth-26-4-billion-by-2024-exclusive-report-by-marketsandmarkets*.

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.

Kshitij Tayal, Rao Nikhil, Saurabh Agarwal, and Karthik Subbian. 2019. Short text classification using graph convolutional network. *NIPS workshop on Graph Representation Learning*.

Arash Vahdat. 2017. Toward robustness against label noise in training deep discriminative neural networks. In *Advances in Neural Information Processing Systems*, pages 5596–5605.

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 International Conference on Computer Vision*, pages 322–330.

Xiang Wu, Ran He, Zhenan Sun, and Tieniu Tan. 2018. A light cnn for deep face representation with noisy labels. *IEEE Transactions on Information Forensics and Security*, 13(11):2884–2896.

Tong Xiao, Tian Xia, Yi Yang, Chang Huang, and Xiaogang Wang. 2015. Learning from massive noisy labeled data for image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2691–2699.

Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Graph convolutional networks for text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7370–7377.

Hsiang-Fu Yu, Chia-Hua Ho, Prakash Arunachalam, Manas Somaiya, and Chih-Jen Lin. 2012. Product title classification versus text classification. *Csie. Ntu. Edu. Tw*, pages 1–25.

Zhilu Zhang and Mert Sabuncu. 2018. Generalized cross entropy loss for training deep neural networks with noisy labels. In *Advances in neural information processing systems*, pages 8778–8788.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. 2016. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. 2017. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*.

Peng Zhou, Zhenyu Qi, Suncong Zheng, Jiaming Xu, Hongyun Bao, and Bo Xu. 2016. Text classification improved by integrating bidirectional lstm with two-dimensional max pooling. *arXiv preprint arXiv:1611.06639*.