Applying Model-agnostic Methods to Handle Inherent Noise in Large Scale Text Classification

Kshitij Tayal
Department of Computer Science
University of Minnesota, Twin Cities
tayal007@umn.edu

Rahul Ghosh
Department of Computer Science
University of Minnesota, Twin Cities
ghosh128@umn.edu

Vipin Kumar
Department of Computer Science
University of Minnesota, Twin Cities
kumar001@umn.edu

ABSTRACT

Text classification is a fundamental problem, and recently, deep neural networks (DNNs) have shown promising results in many natural language tasks. However, their human-level performance relies on high-quality manual annotations, which are time-consuming and very expensive to collect. In the age of big data, as we move towards large inexpensive datasets, the inherent label noise degrades the generalization of DNNs. While most machine learning literature focuses on building complex networks to handle noise, very few works have studied the performance of simpler methods that can give a swift impact. 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 label smoothing regularization, mixup, and various noise-robust loss functions on three datasets encompassing three popular classification models, i.e., feed forward neural networks, convolutional neural networks, long short-term memory. 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 learnings and demonstrate the application of our approach, which outperformed baselines by up to 10 % in classification accuracy while requiring no network modifications.

KEYWORDS

Text Classification, Natural Language Processing, e-Commerce, Noisy label, Model-agnostic methods

ACM Reference Format:
Kshitij Tayal, Rahul Ghosh, and Vipin Kumar. 2018. Applying Model-agnostic Methods to Handle Inherent Noise in Large Scale Text Classification. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/1122445.1122456

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

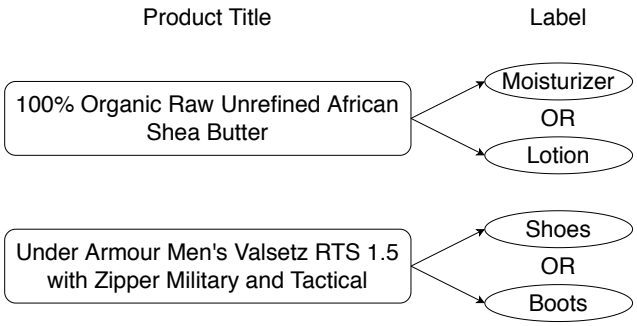

Figure 1: Noisy labels arising due to labels assigned by multiple annotators. Many such classes have subtle differences which creates confusion leading to noisy labels.

domains, such as social media [16], cognitive-biometric recognition [31] ,e-commerce [34, 44] and healthcare [30]. 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 [36] and substantially improve the customer experience. For example, in an e-commerce platform, one might be interested in understanding customer queries to recommend and promote the relevant products. Similarly, automated classification of users into cohorts can result in a higher conversion rate for targeted ads.

Currently, DNNs [4, 17, 49] are the state of the art machine learning models widely deployed for text classification tasks in major enterprises [2, 10, 22]. Like any other supervised classifiers, the performance of these DNNs trained using standard cross-entropy loss is strongly dependent on the quality and quantity of the data. A recent study has demonstrated the memorization capacity of DNNs [1, 45], in which the authors showed that DNNs can completely memorize random labels. This type of behavior significantly harms the generalization performance of DNNs when label noise is present in the training dataset.

Collecting high-quality manual labels is time-consuming and expensive. At the same time, there are separate, less expensive sources to collect labeled data, such as Mechanical Turk [19], search engine meta data, and social media tags. Nevertheless, these inexpensive large datasets introduce noise in the dataset 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 Figure 1. Blindly trusting these large

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/1122445.1122456

inexpensive datasets as gold-standard can decrease the performance of models, and can perturb the optimal learning representations.

Learning from noisy labels is an active area of research in computer vision, and several model cognizant approaches [20, 41] have been proposed. However, these approaches work on building complex networks to handle noise and require substantial background knowledge and training to operate. Conversely, there is minimal research studying the performance of the model-agnostic methods to handle inherent label noise on large scale text classification tasks. 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. Our contribution is to provide insights into the model-agnostic methods to handle inherent noise in text classification. The main advantage of these methods is that they can be easily incorporated into existing machine learning workflows with no network modifications.

Under model agnostic schemes, there are several different lines of work which include modeling noise-transition matrix [8, 28, 37], training auxiliary network [9, 13], training with clean labels [24], label regularization [38], data augmentation [46], and noise-robust loss functions. In large scale text datasets having hundreds of distinct classes, getting a clean dataset is a significant challenge. Also, 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 [23, 32, 40, 48] 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 [13] or has strong assumptions on the type of noise present [37].

We conduct our study on large web-scale text data scraped from popular e-commerce platforms as our training data, and the labels are inferred from user-provided metadata. In contrast with previous work [14, 21], we do not introduce any external noise, and our dataset contains a significant number of classes leading to a higher inherent noise due to annotator confusion. Our research encompasses three popular text classification models i.e., feed forward neural networks (FFNN), convolutional neural networks (CNN), and long short-term memory (LSTM) networks. 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.
- We also show that model-agnostic methods consistently outperform baselines over a wide variety of neural networks such as FFNN, LSTM and CNN.

Code to reproduce the experiments from the paper can be found at the following link - *https://github.com/tayalkshitij/model-agnostic-methods*. The rest of the paper is organized as follows. We describe the problem formulation in Section 3. In Section 3.1, we motivate the specific problems we solve, and Section 3.2 provide details on the datasets we use. In Section 3.3, we give brief details about model-agnostic methods we used in our evaluation, and provide the details about the model architecture in Section 4. In Section 4.3, we compare our methods and provide extensive qualitative and quantitative result. Finally, Section 6 concludes the paper, with possible directions for future research.

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 clean class to noisy class. Mnih et al. [26] 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. [37] 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. [28] 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. [42] developed a noise model, where noise is modeled on the instance and its class. Specifically, the noisy observed annotation is conditioned on the binary random variable, indicating if an instance label is mistaken. Similarly, Vahdat [39] model the noise through Conditional Random Fields (CRF), where the clean labels are modeled as latent variables during training.

Under training auxiliary network, Malach et al. [24] proposed training two different networks which back-propagate the loss when the predictions of the two network disagree. Mentor network [13], 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, [9] presents 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. [27] presented robust surrogate loss functions for handling noisy labels in a binary classification task. Mean absolute error (MAE) [7] was shown to be inherently robust to label noise for the classification task. Similarly bootstrapping loss function [32] was proposed, which introduced a weighted combination of target labels and network predictions to compensate for noisy samples. While [32] uses a fixed hyper-parameter as weights, D2L [23] 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 [48] was proposed, which is a combination of MAE and categorically cross entropy (CCE) loss. Symmetric cross-entropy [40] augments the standard CCE, similar to symmetric KL-divergence, with the noise robust reverse cross-entropy. Data augmentation is another popular procedure popularly used to handle noisy labels. These include mixup [46] that uses convex combinations of training data points and its corresponding labels to make the training procedure more robust to label corruption. Label Smoothing Regularization (LSR) [38] is another method where the 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, in which we predict the categorical label of a given text. Each data instance can be described through the attribute $(\boldsymbol{x}, \boldsymbol{y})$ defined as follows

- Features $\boldsymbol{x} \in \mathbb{R}^d$ is the vector representation of a text, where d is the dimensionality of the embedding vector. This representation can be obtained by a bag-of-words approach [47] or by more complex model such as a word2vec [25] or fastText [15].
- Label $\boldsymbol{y} \in \{0, 1\}^K$ is the one hot encoded vector (i.e., equivalent to an integer between 1 and K).

Data: We are given a dataset of N samples denoted by

$$\mathcal{D} = \{\boldsymbol{x}_i, \boldsymbol{y}_i\}_{i=1}^N, \tag{3.1}$$

where K $\lll$ N. We split our dataset $\mathcal{D}$ 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 between $\boldsymbol{x}_i$ and corresponding label $\boldsymbol{y}_i$ such that it makes correct prediction on test sample $x_i \in \mathcal{D}^{Test}$. Here θ are the parameters of the DNN models.

3.2 Data Description

We conduct our study on large web-scale product title data scraped from popular e-commerce platforms as our training data. They consist of Amazon product titles and other side information [11][1]. The datasets are broken down by categories, and we make use of three such categories i.e. **Electronics**, **Beauty** and **Automotive**. Table 2 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, it's 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. Each category has $200K - 300K$ samples distributed across $300 - 1800$ classes. Product title is short which is evident from the average length ($10 - 15$ words) found in our dataset. A sample dataset is shown in table 1

Table 1: Sample product title and their category for automotive dataset

Product Title	#Label
Gunk Liquid Wrench 4 Oz	Radiators
Simoniz Car Wash System	Cleaners
Lock De-Icer Keychain	De-Icers
Tuggy T-Post Puller	Puller Sets
Preval 267 Spray Gun	Spray Guns

3.3 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, CCE loss is one such loss function which measures the performance of a classification model whose output is a likelihood estimation $f(\boldsymbol{x}; \theta)$ between 0 to 1 scale. It increases as the predicted likelihood estimation deviates from the actual label and a perfect model will have 0 loss. The CCE loss is given by

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

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.

Minimization of loss function, intuitively, tries to make sure that the prediction does not differ too much from target label. At the beginning of training, the error in prediction is high and gradually reduces with the progress of training. Most modern machine learning models are DNNs which are trained to achieve near zero classification losses in the training data. 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 DNNs the output of the trained model matches the training labels exactly [45]. However, 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.

3.3.1 ***Label Smoothing Regularization*** *[38].* As training progresses, the cross-entropy objective function gets minimised, which is equivalent to the log-likelihood of the correct-label getting maximized. This encourages the model to be more confident on its predictions by minimizing the probabilities of the given class. This 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, Label Smoothing Regularization (LSR) proposes to use a mixture of

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

Dataset	#Samples	#Training	#Test	#Unique Words	#Class	Average Length
Beauty	207574	145302	62272	118215	342	10.28
Electronics	362142	253500	108642	424368	823	14.65
Automotive	243296	170307	72989	228234	1818	10.00

Table 2: Sumary statistics of datasets

the original ground truth distribution with another fixed distribution u in place of the original labels. The target label is modified as follows

$$y'_{ij} = (1-\epsilon)y_{ij} + \epsilon u(j) \tag{3.3}$$

where, $u(j)$ is used as a fixed prior distribution over labels weighted by ϵ. Thus, using this weighted target label, the new regularized loss function takes the form,

$$\mathcal{L}_{LSR} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{K}\left[(1-\epsilon)y_{ij}log(f_j(\boldsymbol{x}_i;\theta)) + \epsilon u(j)log(f_j(\boldsymbol{x}_i;\theta))\right] \tag{3.4}$$

3.3.2 ***Bootstrapping****[32].* 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. Bootstapping loss function dynamically updates the target labels based on the current state of the model. More precisely, the updated target label is 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. The main idea is to focus less on noisy labels in favor of making model predictions more stable as the learning proceeds. This approach is referred to as soft bootstrapping when the predicted probabilities are directly used to generate target labels as follows

$$\mathcal{L}_{boot\text{-}soft} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{K}[\beta y_{ij} + (1-\beta)f_j(\boldsymbol{x}_i;\theta)]log(f_j(\boldsymbol{x}_i;\theta)) \tag{3.5}$$

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

$$\mathcal{L}_{boot\text{-}hard} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{K}[\beta y_{ij} + (1-\beta)z_{ij}]log(z_{ij}) \tag{3.6}$$

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

3.3.3 ***mixup*** *[46].* Mixup is a simple data augmentation technique that works on the vicinal risk minimization principle [3], 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,

$$\tilde{x} = \lambda \boldsymbol{x}_i + (1-\lambda)\boldsymbol{x}_j \tag{3.7}$$

$$\tilde{y} = \lambda \boldsymbol{y}_i + (1-\lambda)\boldsymbol{y}_j \tag{3.8}$$

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. Moreover, the generated labels acts as soft targets making memorization of the noisy labels more difficult to achieve. 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.

3.3.4 ***D2L Loss*** *[23].* The authors found that while training with noisy labels, DNNs transform the data into sub-spaces with lower dimensionality during the initial stages and thereafter progressively attempts to accomodate the noisy labels by increasing the subspace dimensionality. This leads to the overfitting of the network to the noisy labels due to high dimensional decision boundaries. Thus, to avoid the effect of noisy labels, a label smoothing strategy is proposed which finds an optimal trade-off between the prediction performance and the subspace dimensionality. SPecifically, the model is trained with the training labels until a turning point is found where the model starts to overfit. This turning point is found on the basis of Local Intrinsic Dimesnionality (LID), which is a measure of the subspace dimensionality at each epoch. Hereafter the training labels are smoothed by adding the network prediction to them and this smoothed labels are used for training the models according to the loss function shown below

$$\mathcal{L}_{D2L} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{K}[\alpha_t y_{ij} + (1-\alpha_t)f_j(\boldsymbol{x}_i;\theta)]log(f_j(\boldsymbol{x}_i;\theta)) \tag{3.9}$$

where, for t'th training epoch α_t is a LID based weight calculated as,

$$\alpha_t = exp\left(-\lambda\frac{LID_t}{\min_{i=0}^{t} LID_i}\right) \tag{3.10}$$

With the increase in LID score the value of α decreases and thus the model predictions are trusted more instead of the noisy training labels.

3.3.5 ***Generalized Cross Entropy (GCE)*** *[48].* The L_{GCE} loss is a generalization of CCE and mean absolute error (MAE) with hyperparameter q, where $q \in [0, 1]$. When $q \to 0$, L_{GCE} loss becomes CCE, and likewise becomes MAE/unhinged loss when $q = 1$. In the gradient step of CCE, samples with predictions that differ from the target labels are inherently weighted more than the samples where the prediction agrees more with the target label. Consequently,

during training, more stress is put on samples where the model disagrees with the target labels. This implicit weighting scheme 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 [7]. 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 [6, 48]. L_{GCE} loss addressed the challenge by taking advantage of both the noise-robustness provided by MAE and the implicit weighting scheme of CCE. The L_{GCE} loss is given by eq. 3.11.

$$L_{GCE} = \frac{1 - (\sum_{i=1}^{i=N} y_i \hat{y_i})^q}{q}, q \in [0, 1] \tag{3.11}$$

3.3.6 ***Symmetric Cross Entropy (SL)*** *[40].* 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, the authors 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

$$\mathcal{L}_{SL} = \alpha \mathcal{L}_{CCE} + \beta \mathcal{L}_{RCE} \tag{3.12}$$

$$\mathcal{L}_{SL} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{K} \alpha y_{ij} log(f_j(\boldsymbol{x}_i; \theta)) + \beta f_j(\boldsymbol{x}_i; \theta) log(y_{ij}) \tag{3.13}$$

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.

Table 3: Hyperparameters for model-agnostic methods

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

4 EXPERIMENTS AND RESULTS

In this section, we evaluate the use of several model-agnostic approaches i.e., label smoothing regularization, *mixup*, and noise-robust loss functions on several large scale datasets shown in Table 2. We use some of the popularly used text classification models: Convolution Neural Network, Long short-term memory, and feed-forward neural network. In this work, we attempt to answer the following research 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: Feed Forward Neural Network [33] or multi-layer perceptrons are the most simple neural network. In this work, we use average pooling operation [35] on input feature and feed forward architecture with two hidden layers having 1024 and 512 units respectively. We employ a ReLU activation function on hidden layers with 0.2 dropout followed by a output layer of K output values, where K is the number of classes.

CNN: Kim et al. [17] introduced the idea of using a CNN to classify text with the central intuition to see text as images. In this work, we have fixed the maximum length of sentence to 10 and embedding size to 128. In our network architecture, we use one convolutional layer having 128 filters with a convolution window/kernel size of 5, followed by a fully connected layer with 512 neurons. The final layer is the output layer of K output values. We use max-pooling in our convolution layer.

LSTM: LSTMs [12] keep track of arbitrary long-term dependencies in the input sequences, which makes it popular among major technological enterprises. As with CNN, we vectorize each text with a matrix of numbers with the shape $10x128$, where the maximum length of sentence is 10, and embedding size is 128. In our network architecture, the first layer of LSTM is the embedding layer followed by variational dropout. The next layer is the LSTM layer, with 256 memory units, followed by the output layer of K output values.

4.2 Experimental setup

We trained all models for a maximum of 75 epochs using Adam optimizer [18] with 0.001 learning rate and terminate training if the validation loss does not reduce for 10 continuous epochs. All the datasets are split into 70% training and 30% testing where further 20% from the training set was kept aside for validation. 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 for each model type. Individual words are encoded using glove embeddings [29] before feeding it into the models. For electronics dataset, we fixed hyperparameter for each of the methods using grid search based on their average performance by 5 fold cross-validation. Due to constraints in the use of computational hardware, we fixed the same parameter for other datasets too. The value of the hyperparameters used are presented in Table 3.

4.3 Results

Table 4 reports the relative performance of the different model-agnostic methods. The column for *CCE* shows the absolute baseline

MODEL	DATASET	*CCE*(%)	*LSR*	*Boot-hard*	*Boot-soft*	*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

Table 4: Relative performance of different model-agnostic methods against cross-entropy loss with no external noise

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:

- All values in result table 4 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.
- *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.
- Weaker methods such as CNN achieved a higher gain in performance.

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. More specifically, *D2L* provide an accuracy increase of 1.5 % - 10 %. 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*. More precisely, *GCE* provides an accuracy increase of 1.5 % - 9.5 %. This shows the benefit of using the noise-robustness feature of MAE in conjunction with *CCE*. Likewise, *SL* uses a combination of reverse cross-entropy, which adds value when used in conjunction with *CCE*.

mixup is a simple data augmentation technique that 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 smoothening effect of the approach, which relies on the addition of a fixed uniform label distribution to the one-hot labels.

Boot-hard and *Boot-soft* performed fine, but not as high as other methods. We attribute this not so great performance on 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.

4.4 Accuracy Curves

Figure 2 denotes training and test accuracies at every epoch attained by FFNN on the beauty dataset using best-performing methods. We find from the plot of the train and test accuracy that the classifier trained using *CCE* first learned 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 [1, 45] 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.

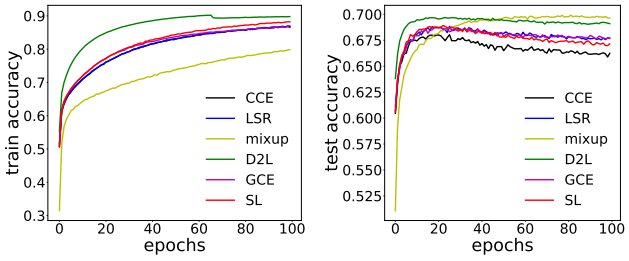

Figure 2: Train and Test accuracy against number of epochs for FFNN on Beauty dataset

MODEL	DATASET	*CCE*(%)	*LSR*	*Boot-hard*	*Boot-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

Table 5: Relative performance of different model-agnostic methods against cross-entropy loss with 20% noise

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 kept intact. Table 5 reports the relative performance of the methods when external noise is added in the datasets. 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. We note that *D2L* gave impressive performance gain over *CCE*. 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-hard* loss function breaks down when we increase the noise, particularly when used with LSTM and CNN in the Electronics dataset. *GCE* and *mixup* also show an increase in relative performance as compared to *CCE*, but is still less than *D2L*.

We further experimented by flipping 40% training labels and find that most of the approaches gave an inferior performance as compared to *CCE*, with the exception of *D2L* and *GCE*. However, the performance gain given by *D2L* and *GCE* was not significant, and we conclude that for text classification with hundreds of labels, model-agnostic methods do not perform well with very large noise.

4.6 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 4, 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 5 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 from these observations, we conclude that the use of model-agnostic methods becomes more necessary when training machine learning models on large datasets with hundreds and thousands of categories.

5 DISCUSSION

While most of the machine learning literature focused on building complex networks to handle noise, very few works [5, 6] have studied the performance of simpler methods that can give a significant impact. In particular, for text classification, to the best of our knowledge, this is the first attempt to apply model agnostic methods requiring no network modifications to handle inherent noise in large scale datasets. 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, having thousands of datapoints and hundreds of labels. We did not find any public dataset that matches our specification, but the major tech enterprises has vast corpora of internal datasets that can benefit from this study. The methods mentioned are easy to implement and can be easily integrated into any machine learning workflows without breaking the existing codebase. *LSR* and *mixup* can be appended as simple functions in the data loader, and other noise-robust loss function can be implemented as a custom loss function with a few lines of code with no computational overhead. In contrast to previous works, we did not add noise and hypothesize that large dataset has thousands of samples and hundreds of unique classes that can inadvertently introduce noise.

We have shown performance gain for popular text classification models, namely CNN, LSTM, and FFNN. However, these methods are general and can be applied in other domains where there are hundreds of classes and a high chance of confusion between some of them. For example, in remote sensing, while annotating satellite image pixels, people frequently get confused between herbs and shrubs class. Also, it is yet to see how these model-agnostic approaches will work for upcoming language models like BERT [4], XLNET [43] having hundreds of millions of parameter. We leave for future work to explore the performance of such a model using model agnostic approaches.

6 CONCLUSION

Deep learning requires a vast amount of training data because of the large number of parameters needed to be tuned by a learning algorithm. Labeled data is crucial to modern machine learning

algorithms, and the manual gold standard dataset is very costly and a limiting factor for many starting enterprises. In this study, we demonstrate the effectiveness of model-agnostic methods in advancing the performance of machine learning models for large scale text classifications. For real-world problems, having large text datasets, we propose to use these methods instead of standard cross-entropy. Specifically, the top-performing approaches *D2L*, *GCE*, *mixup* demonstrates the tangible impact in accuracy over the *CCE* baseline. We fill the gap in existing literature, where applying these methods to large scale text classification tasks is not the norm. 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.

REFERENCES

[1] Devansh Arpit, Stanisław Jastrzębski, 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*. JMLR. org, 233–242.

[2] 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*. 1743–1751.

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

[4] 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* (2018).

[5] Eduardo Fonseca, Frederic Font, and Xavier Serra. 2019. Model-agnostic approaches to handling noisy labels when training sound event classifiers. In *2019 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, 16–20.

[6] 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)*. IEEE, 21–25.

[7] 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*.

[8] Jacob Goldberger and Ehud Ben-Reuven. 2016. Training deep neural-networks using a noise adaptation layer. (2016).

[9] 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)*. 135–150.

[10] 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*. 1927–1935.

[11] 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*. International World Wide Web Conferences Steering Committee, 507–517.

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

[13] 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* (2017).

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

[15] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759* (2016).

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

[17] Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882* (2014).

[18] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[19] 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*. 453–456.

[20] 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*. 3204–3213.

[21] 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*. 5051–5059.

[22] 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*. 1831–1841.

[23] 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* (2018).

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

[25] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. 3111–3119.

[26] 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)*. 567–574.

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

[28] 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*. 1944–1952.

[29] 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)*. 1532–1543.

[30] John P Pestian, Christopher Brew, Paweł Matykiewicz, Dj J Hovermale, Neil Johnson, K Bretonnel Cohen, and Włodzisław Duch. 2007. A shared task involving multi-label classification of clinical free text. In *Proceedings of the Workshop on BioNLP 2007: Biological, Translational, and Clinical Language Processing*. Association for Computational Linguistics, 97–104.

[31] 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 (2016), 134–143.

[32] 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* (2014).

[33] 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.

[34] Dou Shen, Ying Li, Xiao Li, and Dengyong Zhou. 2009. Product query classification. In *Proceedings of the 18th ACM conference on Information and knowledge management*. ACM, 741–750.

[35] 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* (2018).

[36] 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* (2019).

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

[38] 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*. 2818–2826.

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

[40] 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*. 322–330.

[41] 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 (2018), 2884–2896.

[42] 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. 2691–2699.

[43] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. 2019. XLNet: Generalized Autoregressive Pretraining for Language Understanding. *arXiv preprint arXiv:1906.08237* (2019).

[44] 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* (2012), 1–25.

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

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

[47] Yin Zhang, Rong Jin, and Zhi-Hua Zhou. 2010. Understanding bag-of-words model: a statistical framework. *International Journal of Machine Learning and Cybernetics* 1, 1-4 (2010), 43–52.

[48] 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*. 8778–8788.

[49] 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* (2016).