

# Towards Robustness to Label Noise in Text Classification via Noise Modeling

Siddhant Garg\*, Goutham Ramakrishnan\*, Varun Thumbe\*

Amazon Alexa AI , Health at Scale , KLA Corporation



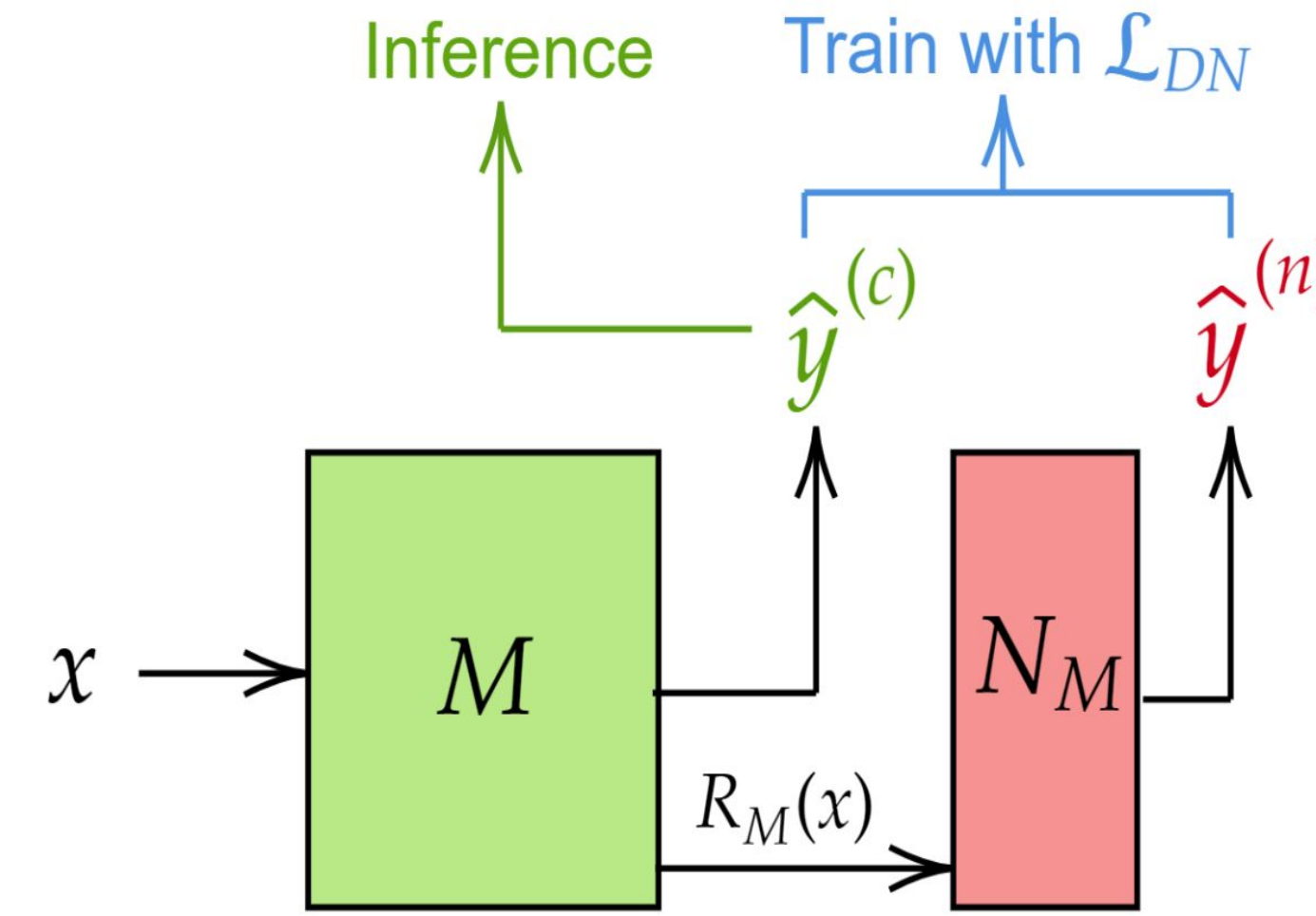
## Motivation

- Obtaining large scale noise-free datasets for text classification is very challenging and expensive
- Crowd sourced datasets, from platforms like MTurks, have inevitable human annotation errors due to:
  - Ambiguity of annotations
  - Inexperience of annotators
  - Human error due to annotation speed
- Label noise in samples can be of the following form:
  - Random (Randomly assigning a label to sample)
  - Label-dependent (Confusing a specific label x for y)
  - Input instance-dependent
- Learning with noisy labels is extensively explored for CV tasks, but not for NLP tasks (cannot directly apply CV techniques due to discrete nature of input space)

## Related Work

- Noisy labels for NLP tasks [Jindal et al 2019 NAACL]
  - Learn a label dependent *noise model* (probability matrix) over the classifier model
  - Use  $l_2$ -regularizer on the noise model weights with no selective guiding for learning noisy samples
- Mixture Models for Noisy and Clean labels in CV [Arazo et al 2019 ICML]
  - Learning from clean labels is easier than learning from noisy labels initially
  - Training loss in early epochs clusters into 2 regions corresponding to samples with clean and noisy labels
  - A mixture model(Beta/Gaussian) can be fit to get the probability of sample label being clean or noisy

## 3-Step Training Methodology



1. **Warmup:** Train the classifier **M** for some warmup epochs( $T_0$ ) by minimizing the CE loss between  $\hat{y}^{(c)}$  (predicted clean output) and  $y'$ (noisy ground-truth)
2. **Fit BMM:** Fit a Beta Mixture Model  $\mathcal{B}(x)$  on the CE loss( $\hat{y}^{(c)}, y'$ ) distribution after warmup to estimate probability of sample having noisy or clean labels
3. **Train M and  $N_M$ :** Use probability scores from the fitted BMM with the de-noising loss to train end-to-end.

## De-noising Loss Formulation

$$\mathcal{L}_{DN-S} = \mathcal{L}_{CE}(\hat{y}^{(n)}, y) + \beta \cdot \mathcal{B}(x) \cdot \mathcal{L}_{CE}(\hat{y}^{(c)}, y)$$

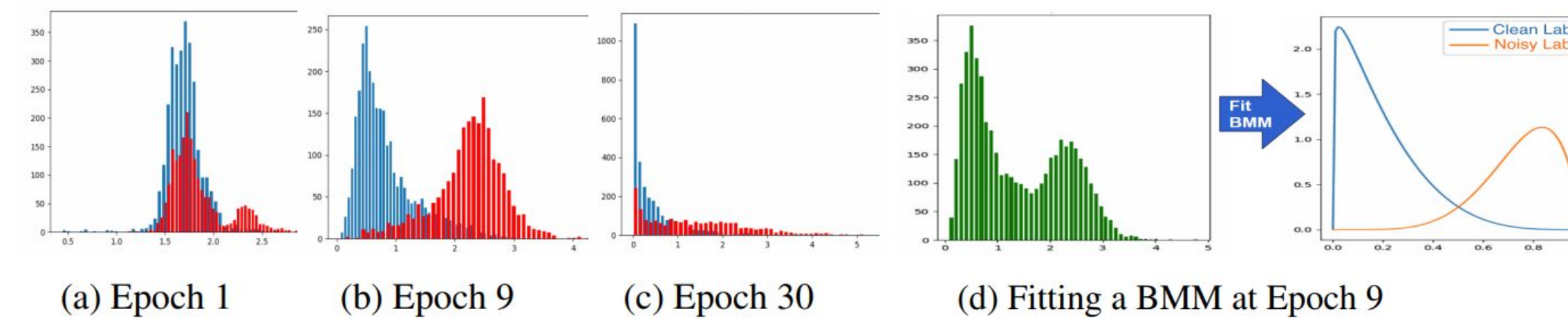
Soft Denoising Loss

Trains cascade  $M-N_M$

$$\mathcal{L}_{DN-H} = \mathcal{L}_{CE}(\hat{y}^{(n)}, y) + \beta \cdot \mathbb{1}[\mathcal{B}(x) > 0.5] \cdot \mathcal{L}_{CE}(\hat{y}^{(c)}, y)$$

Hard Denoising Loss

Train Model **M**



## Experiments and Results

- Datasets: TREC and AG-News
- Model **M**: 2-layer LSTM, word-CNN with GloVe embed.
- Noise Model  $N_M$ : 2-layer feedforward NN over logits from **M**

**Random Noise:** Pick a random % of samples (noise %) and randomly assign them one of the class labels

TREC dataset (100 epochs)						AG_NEWS dataset (30 epochs)							
Noise Percent	Baseline		$\mathcal{L}_{DN-H}$		$\mathcal{L}_{DN-S}$		Noise Percent	Baseline		$\mathcal{L}_{DN-H}$		$\mathcal{L}_{DN-S}$	
	Best	Last	Best	Last	Best	Last		Best	Last	Best	Last	Best	Last
0	93.8	93.0	94	92.6	95	94	0	92.5	92.05	92.38	92.01	92.75	92.61
10	88	88.6	92.2	91.6	92.4	91.4	10	91.9	90.2	91.5	91.4	91.8	91.5
20	89.4	79.8	90.2	90	90	90.2	20	91.3	89.75	90.58	90.79	90.76	91.02
30	83.4	72.4	88.8	88.4	87.4	85.4	30	90.5	87.97	90.82	90.9	91	90.9
40	79.6	54.8	83	79.4	83.4	82.4	40	89.31	85.57	90.35	90.36	90.29	90.15
50	77.6	50.4	82.4	82.4	82.6	74.2	50	88.63	78.1	89.3	88.9	88.57	88.47

**Input-Dependent Noise (TREC):** Two types of label-noise-  
 1) Samples starting with “How”/“What”: Insert random noise  
 2) Randomly flip labels for the longest x% of samples

Noise inserted randomly for text starting with “How” or “What “						Label noise added to the longest x% of inputs							
Noise Percent	Baseline		$\mathcal{L}_{DN-H}$		$\mathcal{L}_{DN-S}$		Noise Percent	Baseline		$\mathcal{L}_{DN-H}$		$\mathcal{L}_{DN-S}$	
	Best	Last	Best	Last	Best	Last		Best	Last	Best	Last	Best	Last
0	93.8	93.0	94	92.6	95	94	0	93.8	93.0	94	92.6	95	94
10	89.2	88.8	91.8	91.8	91.8	92	10	91.4	90.4	91.6	91	92	92.4
20	84.4	76.2	87.4	85.2	90.6	89.4	20	87	87.6	90.2	89.4	90.6	91.6
30	77.8	67.2	84.2	84.6	83.8	77	30	82.2	84	87.4	87.2	85.4	85.6
40	76	59	79	80	79.2	60	40	82.4	79.8	87.4	86.6	84	81.2
50	71.8	56	67.8	69.2	75.6	59.8	50	74.2	71.2	79	79	75	72

**Robustness to over-fitting on label noise:** Observe test loss on increasing training epochs on TREC dataset at different %-random noise levels

