Noise Learning for Text Classification: A Benchmark

Bo Liu
School of Sofrware Engineering, Xi'an Jiaotong University
the CETC Key Laboratory of Smart City Model Simulation and Intelligent Technology
The Smart City Research Institute of CETC and National Center for Applied Mathematics Shenzhen(NCAMS)
boliu@stu.xjtu.edu.cn

Wandi Xu
Northeastern University
xuwandi@stumail.neu.edu.cn

Yuejia Xiang
Tencent
yuejiaxiang@tencent.com

Xiaojun Wu
the CETC Key Laboratory of Smart City Model Simulation and Intelligent Technology
The Smart City Research Institute of CETC and National Center for Applied Mathematics Shenzhen(NCAMS)
wuxiaojun@cetc.com.cn

Lejian He
Cornell University
lh628@cornell.edu

Bowen Zhang
Shenzhen Technology University
zhang_bo_wen@foxmail.com

Li Zhu*
School of Sofrware Engineering, Xi'an Jiaotong University
zhuli@xjtu.edu.cn

Abstract

Noise Learning is important in the task of text classification which depends on massive labeled data that could be error-prone. However, we find that noise learning in text classification is relatively underdeveloped: 1. many methods that have been proven effective in the image domain are not explored in text classification, 2. it is difficult to conduct a fair comparison between previous studies as they do experiments in different noise settings. In this work, we adapt four state-of-the-art methods of noise learning from the image domain to text classification. Moreover, we conduct comprehensive experiments on our benchmark of noise learning with seven commonly-used methods, four datasets, and five noise modes. Additionally, most previous works are based on an implicit hypothesis that the commonly-used datasets such as TREC, Ag-News and Chnsenticorp contain no errors. However, these datasets indeed contain 0.61% to 15.77% noise labels which we define as **intrinsic noise** that can cause inaccurate evaluation. Therefore, we build a new dataset Golden-Chnsenticorp (G-Chnsenticorp) without **intrinsic noise** to more accurately compare the effects of different noise learning methods. To the best of our knowledge, this is the first benchmark of noise learning for text classification.

* Corresponding author.

1 Introduction

The fast development of text classification cannot be achieved without massive labeled data resources, especially for supervised embedding-based methods. However, not all training data are correctly labeled in practice (Wang et al., 2018; Zlateski et al., 2018). These incorrectly labeled data are called noisy labels. To alleviate the interference caused by noisy labels, many noise learning methods have been proposed (Rolnick et al., 2017; Veit et al., 2017; Jiang et al., 2018; Yang Liu, 2019; Li et al., 2020; Curtis G. Northcutt, 2020; Garg et al., 2021). Although both CV and NLP domains have serious label noise problems, these work are mainly focused on the CV domain, only (Garg et al., 2021) is dedicated to NLP domain. So in order to support the development of noise learning in NLP, we would like to propose a noise learning benchmark in the field of text classification.

We find that the previous studies in noise learning for text classification tasks have two weaknesses. 1. The implicit hypothesis is unreasonable. 2. They lack horizontal comparison.

Unreasonable implicit hypothesis. The previous research uses a four-step approach to evaluate a new method. First, they split a dataset into training data and test data, and then add manufactured noise data to the training data following a predefined noise mode. Third, they apply the noise learning method to the training data. In the end, they evaluate the noise learning method on the test data. This approach makes an implicit assumption that the dataset is completely reliable. Based on this assumption, the noise data in the experiment is equal to

Proceedings of the 29th International Conference on Computational Linguistics, pages 4557–4567
October 12–17, 2022.

Dataset	Intrinsic Noise			Total
	Fatal	Inexact	Ambiguous	
TREC	1.94%	2.58%	3.16%	7.68%
Ag-News	0.00%	0.12%	0.49%	0.61%
Chn.	2.19%	4.63%	8.95%	15.77%

Table 1: The ratio of **intrinsic noise** in several widely used datasets. Chn. denotes Chnsenticorp.

the manufactured noise data and the evaluation given the test data is accurate. But we find that the dataset is not completely noise-free and the ratio of noise data in the dataset (i.e. **intrinsic noise**, which can be divided into three parts according to the ambiguity level: fatal noise, inexact noise, and ambiguous noise) is not negligible for noise learning task (Han et al., 2020), as shown in Table 1. Thus, the noise data in the experiment is actually equal to the superposition of the **intrinsic noise** data and the manufactured noise. Hence, the evaluation in previous studies is not robust and accurate.

Lacking horizontal comparison. The previous studies lack horizontal comparison between them as they usually use different datasets, different noise modes, different noise ratios, etc. This is not conducive to the development of noise learning for text classification.

In order to overcome these two weaknesses, we build a new dataset without **intrinsic noise** and present a benchmark of noise learning for text classification. The main contributions of this paper can be summarized as follows:

- We divide the **intrinsic noise** into three parts according to the ambiguity level. To the best of our knowledge, this is the first time **intrinsic noise** to be defined and analyzed in the noise learning task of text classification.
- We propose a new dataset without **intrinsic noise**, named G-Chnsenticorp. Experiments on this dataset would have more accurate results.
- This is the first time that a benchmark of noise learning for text classification has been established. First, we summarize the noise modes mentioned in previous works. Second, we reproduce/transform seven commonly used noise learning methods in/to the text classification task.
- We have several interesting observations and conclusions: **Intrinsic Noise** is more difficult to be learned than other noise modes; many methods do not work well when the noise ratio is higher than 30%; a small amount of white noise can benefit classification methods, etc.

2 Related Work

Plenty of previous studies have examined the factors that impact label noise learning models. Zhang et al. (2016) prove that a model of sufficient complexity can over-fit any noise. Jacot et al. (2018) analyze convergence and generalization in neural networks from the perspective of Gaussian processes in the infinite-width limit. Rolnick et al. (2017) propose a model that is extremely adaptive to specific patterns of artificial noise. Li et al. (2020) prove that gradient descent with early stopping is robust to label noise for overparameterized neural networks. From the perspective of noise mode, Algan and Ulusoy (2020) conduct a detailed analysis of the influence of label noise on model training and propose a generic framework to generate feature-dependent label noise. Hataya and Nakayama (2018) investigate the behavior of Convolutional Neural Networks (CNNs) under class-dependently simulated label noise. Flatow and Penner (2017) test the robustness of the model by randomly permuting the labels of the training set with increasing frequency.Jiang et al. (2020) establish the first benchmark of controlled real-world label noise in the CV field.

Following the work of Han et al. (2020) which divides noise learning methods into three categories optimization-based method, objective-based method, and data-based method for a more comprehensive comparative analysis, we select several commonly-used noise learning methods from each category to conduct comparison experiments.

There exist a few benchmarks (Xu et al., 2018; Jiang et al., 2020) in the field of image classification, but there is no benchmark in the field of text classification. Therefore, many works in text classification task (Jindal et al., 2019; Garg et al., 2021) only do comparative experiments with their own baseline. Moreover, very few studies consider different noise settings. A robust benchmark is much needed in the development of the field of text classification.

3 G-Chnsenticorp database

Ambiguity level. We then define three categories of intrinsic noise based on the aforementioned annotator agreement thresholds: fatal noise [90%, 100%], inexact noise [60%, 90%), and ambiguous noise [0%, 60%). Specifically, when more than 90% of the annotators agree upon a label different from the original one, we consider the sample as fatal noise. When 60% to 90% of annotators agree on a label different the original, we consider the sample as inexact noise. When less than 60% of annotators agree on the label, no matter what label it is we consider it as ambiguous. Ratios of the three intrinsic noises in TREC, Ag-News and Chnsenticorp are summarized in Table 1. We adopted two annotator agreement thresholds 90% and 60% as our guidance. (Please refer to Appendix B for the selection of threshold.) Note that annotator agreement threshold here means the ratio of agreement on labeling among the annotators.

We investigate three common datasets for text classification illustrated as follows:

- TREC (Voorhees and Tice., 1999): An question

classification dataset consisting of fact-based questions divided into broad semantic categories. There are six classes. It contains 5k+ training samples.

- Ag-News (Xiang Zhang, 2015): A large-scale, four-class topic classification dataset. It contains approximately 110K training samples.

- Chnsenticorp (Tan and Zhang, 2008): A hotel review classification dataset. It contains 5K+ positive reviews and 2K+ negative reviews.

With regard to the Chnsenticorp dataset, we recruited a team of 10 experts in hotel management as annotators to label samples from Chnsenticorp and found that there existed four categories in the dataset: positive, negative, irrelevant, and neutral. We remove irrelevant and neutral samples to construct a new binary dataset Golden-Chnsenticorp (G-Chnsenticorp) since we believe those samples are ambiguous in the task of text classification.

Based on the three types of intrinsic noise, we reconstruct the G-Chnsenticorp dataset as follows: we correct samples of fatal noise with the annotator majority label and remove samples of inexact or ambiguous noise. Note that since G-Chnsenticorp is a simple binary text sentiment classification dataset, we are confident to correct the fatal noise labels with our expert majority label and remove the other two types of noise. Ambiguous noise samples such as "Good breakfast but bad bed" are indeed noise to the binary classification dataset and challenging samples such as sarcastic reviews are not classified as ambiguous or inexact noise during annotating. The new dataset can ensure the robustness of the models trained on it. Through our annotating and modification as mentioned above, we obtain the final G-Chnsenticorp dataset which contains around 4,000 training samples.

4 Noise Generation Methods

In single-label text classification tasks, we assign a corresponding label to each sentence. For all n samples with k different types of classes, we let $D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$ as noise-free dataset, where x_i denotes the ith sentence in the dataset, $y_i \in \{1, \ldots, k\}$ denotes the class of the ith sentence. However, it is hard for us to find truly noise-free data in the dataset except for manual verification. Therefore, we first assume all pre-given data are true. Then, we use noise transfer matrix to automatically generate the corresponding noise-labeled dataset. Here, we set $D^{'} = \{(x_1, y_1^{'}), (x_2, y_2^{'}), \ldots, (x_n, y_n^{'})\}$ as the noisy dataset, where $y_i^{'}$ denotes the corresponding noise label of sentence x_i. The noise transfer matrix $\Phi(y, y^{'})$ represents the transfer distribution of the true label y and the noise label $y^{'}$, which is a $k \times k$ matrix.

Under the same assumption in other studies (Aritra Ghosh, 2017; Patrini et al., 2017; Jindal et al., 2019), the noise label $y_i^{'}$ only depends on the corresponding true label y_i, but not the input x_i or the other labels y_j or $y_j^{'}$. In our experiment, we use the noise transfer matrix Φ to generate the corresponding noise labels for the training set, but labels in the test set are not changed. Meanwhile, we use p to denote the noise rate, which is the overall probability of label errors, where $0 \leq p \leq 1$.

Noise Mode	Noise Generation Formula
Symmetric Noise	$\Phi = (1-p)I + \frac{p}{k}A$
Pairflip Noise	$\Phi = (1-p)I + pB$
Uniform Noise	$\Phi = (1-p)I + \frac{p}{k}C$
Random Noise	$\Phi = (1-p)I + pD$
White Noise	other unrelated field text

Table 2: Different generation formala of noise mode. Here, I represents the identity matrix. A denotes an all-ones matrix with zeros along the diagonal. B represents the identity matrix where the last column is transferred to the first column. C represents the matrix with zeros along the diagonal, and except for the diagonal, the values are uniformly and independently distributed. D is a matrix independent of the $k-1$ dimensional unit simplex with zeros along the diagonal.

Generally speaking, noise labels can be categorized into four types according to different noise transfer matrices. As shown in Table 2 and Figure 1: (1) Symmetric Noise (Van Rooyen et al., 2015); (2) Pairflip Noise (Han et al., 2018); (3) Random Noise (Garg et al., 2021); (4)Uniform Noise (Garg et al., 2021). Here, if i means the original category and j means the convered category, Φ[i][j] represents the probability of class i becoming to class j. Additionally, we define a new type of noise named White Noise. Referring to the practice in the image field (Rolnick et al., 2017), we first collect different fields of text data and generate white noise by randomly labeling the labels.

80%	6.7%	6.7%	6.7%
6.7%	80%	6.7%	6.7%
6.7%	6.7%	80%	6.7%
6.7%	6.7%	6.7%	80%

(a) Symmetric Noise

80%	20%	0%	0%
0%	80%	20%	0%
0%	0%	80%	20%
20%	0%	0%	80%

(b) Pairflip Noise

80%	12.9%	1.7%	5.4%
2.7%	80%	11.1%	6.2%
13.6%	5.0%	80%	1.4%
13.8%	0.1%	6.1%	80%

(c) Random Noise

80%	3.8%	10.1%	6.1%
0.5%	80%	2.6%	16.9%
6.7%	5.6%	80%	7.7%
8.2%	8.7%	3.1%	80%

(d) Uniform Noise

Figure 1: Transition matrices of different noise types (using 4 classes and noise rate p=0.2 as an example).

5 Method

Common noise learning methods can be divided into three categories: optimization-based method, objective-

based method and data-based method (Han et al., 2020).

5.1 Optimization-based Method

Optimization-based methods use two networks to make predictions on the same mini-batch data and calculate a joint loss with Co-Regularization for each training example. For the optimization policy, the key is to explore the dynamic process of optimization, which relates to memorization. (Han et al., 2020)

Here, we modify the Co-teaching (Han et al., 2018), Co-teaching+ (Yu et al., 2019) and JoCoR (Wei et al., 2020) frameworks to make the data and models compatible with natural language processing tasks. All of them are originally used in the field of computer vision.

- Co-teaching: it trains two networks simultaneously. In each batch data, both networks select their small-loss samples to cross-update parameters of the other network.
- Co-teaching+: it trains two networks simultaneously, too. Different from selecting all small-loss data in Co-teaching, Co-teaching+ only keeps prediction discrepancy data in the two networks.
- JoCoR: it also trains two networks, but updates parameters with a joint loss. To reduce divergence between two networks, JoCoR uses the joint loss and sampling discrepancy data to backward propagate in a whole.

For feature representation layers, we use three types of networks to extract features, which are FNN, CNN, and BERT.

- Co-teaching$_{FNN}$: based on the Co-teaching method, we use three-layer feedforward neural networks to extract features.
- Co-teaching$_{CNN}$: based on the Co-teaching method, we use CNN for feature representation.
- Co-teaching$_{BERT}$: based on the Co-teaching method, we use BERT for feature extraction.

5.2 Objective-based Method

The objective-based methods learn from noisy data by modifying the objective function. Specifically, the key is to design a suitably modified loss, which is noise-tolerant and guarantees statistical consistency compared to the original loss (Han et al., 2020).

Here we select LSTM$_{DN-H}$ (Garg et al., 2021), LSTM$_{DN-S}$ (Garg et al., 2021) and Peer (Yang Liu, 2019) to modify the loss function. In these three approaches, only Peer is originally used in the field of natural language processing.

- LSTM$_{DN-H}$: the network first assigns a probability score to each training data by a beta mixture model clustering the losses at an early epoch of training. Then the network is trained with these scores using the joint loss l_{DN-H}.

$$l_{DN-H} = l_{CE}(\hat{y}^{(n)}, y) + \beta \cdot B(x) \cdot l_{CE}(\hat{y}^{(c)}, y)$$

where $\hat{y}^{(n)}$ denotes the noisy label prediction, $\hat{y}^{(c)}$ denotes the clean label prediction, y denotes the input in training dataset, l_{CE} denotes the cross entropy loss, B(x) denotes the posterior probability that x has a clean label, and β is a weighting parameter between the two terms.

- LSTM$_{DN-S}$: similar to the LSTM$_{DN-H}$. The only difference is that the network uses an alternative formulation by replacing the Bernoulli R.V. B(x) with the indicator $\mathbb{1}[B(x) > 0.5]$. The l_{DN-S} loss function is as follows:

$$l_{DN-S} = l_{CE}(\hat{y}^{(n)}, y) + \beta \cdot \chi \cdot l_{CE}(\hat{y}^{(c)}, y)$$

where χ denotes the indicator $\mathbb{1}[B(x) > 0.5]$. In the experiments of LSTM$_{DN-H}$ and LSTM$_{DN-S}$ model,we use pretrained word2vec embeddings and lstm neural network layer to extract features.

- Peer: introduces a new family of loss functions called peer loss functions.This method enables training a classifier over noisy labels without using explicit knowledge of the noise rates of labels. The l_{peer} loss function is as follows:

$$l_{peer}(f(x_j), y_j) = l_1(f(x_j), y_j) - \alpha l_2(f(x_{j_1}), y_{j_2})$$

where $alpha$ is non-zero real number hyper-parameter. For each sample (x_j, y_j), randomly draw other two samples (x_{j_1}, y_{j_1}), (x_{j_2}, y_{j_2}) such that $j_1 \neq j_2$. These two samples are called the peer sample. f represents the bayes optimal classifier. l_1 and l_2 can be any standard classification-calibrated loss function, such as cross entropy loss and mean square error loss. In our experiments, we use BERT (Devlin et al., 2019) as the feature extraction layer.

5.3 Data-based Method

For data-based methods, we aim to discover the underlying noise transition pattern. The noise transfer matrix allows us to find the relationship between the clean label and the noisy label. Therefore, the key point here is to design an accurate estimator of the noise transition matrix. In this method, we select the Confident learning approach that is originally used in computer vision to solve this problem.

Confident learning (CL) is an alternative approach that instead focuses on the label quality by characterizing and identifying label errors in datasets. Based on the principles of pruning noisy data, we count with probabilistic thresholds to estimate noise and rank examples to train with confidence (Curtis G. Northcutt, 2019). The CL model inferred which samples are noisy by obtaining the predicted probabilities of the samples on different classifications. In other words, the predicted probabilities are the feature of the CL model to discriminate noisy labels. In our experiments,we use three-layer feedforward neural networks to extract predicted probabilities of the samples.

			Symmetric Noise						Pairflip Noise					
	Method	Clean Data	10%	20%	30%	40%	50%	60%	10%	20%	30%	40%	50%	60%
	Co-teaching	88.40%	85.00%	82.20%	80.80%	73.80%	66.80%	64.00%	85.00%	79.60%	74.40%	53.60%	33.00%	15.80%
	Co-teaching+	84.60%	84.00%	82.40%	81.20%	78.80%	74.20%	68.40%	83.80%	84.00%	77.40%	72.40%	46.60%	29.40%
	JoCoR	84.80%	83.20%	80.20%	80.80%	77.60%	67.20%	63.80%	84.40%	78.40%	75.80%	48.20%	31.00%	28.60%
	LSTM$_{DN-H}$	94.20%	**92.20%**	89.20%	85.80%	83.30%	82.40%	81.10%	91.90%	88.80%	85.60%	84.50%	83.00%	**81.50%**
	LSTM$_{DN-S}$	**94.40%**	**92.20%**	**90.70%**	**87.80%**	**84.80%**	**83.20%**	**82.00%**	**92.10%**	**90.20%**	**88.30%**	**86.30%**	**83.40%**	**81.50%**
	Peer	78.44%	77.52%	75.03%	73.84%	73.11%	71.33%	64.89%	76.99%	75.38%	75.14%	73.03%	66.80%	27.41%
	CL	82.63%	83.57%	84.57%	81.69%	77.35%	70.74%	61.32%	83.17%	77.56%	72.34%	54.11%	30.26%	22.04%
			White Noise						Random Noise					
	Method	Clean Data	10%	20%	30%	40%	50%	60%	10%	20%	30%	40%	50%	60%
	Co-teaching	88.40%	85.80%	86.40%	87.40%	85.40%	83.00%	81.40%	83.80%	81.80%	75.80%	70.40%	67.60%	47.80%
	Co-teaching+	84.60%	83.00%	84.20%	84.40%	82.60%	82.20%	83.40%	83.20%	82.80%	80.80%	72.40%	69.80%	60.60%
	JoCoR	84.80%	85.40%	85.60%	82.60%	82.60%	82.00%	82.80%	84.40%	82.60%	75.60%	72.80%	69.20%	63.00%
	LSTM$_{DN-H}$	94.20%	94.30%	**94.40%**	**94.30%**	**94.00%**	93.50%	**93.80%**	92.00%	89.60%	86.50%	83.40%	**82.00%**	**81.40%**
	LSTM$_{DN-S}$	**94.40%**	**94.40%**	94.20%	94.20%	93.80%	**93.80%**	93.60%	**92.20%**	**91.10%**	**88.80%**	**83.50%**	81.80%	**81.40%**
	Peer	78.44%	77.15%	77.35%	76.93%	75.21%	76.60%	76.73%	74.74%	71.23%	70.05%	68.02%	65.31%	58.98%
	CL	82.63%	83.15%	83.17%	81.74%	74.80%	71.18%	65.36%	83.37%	80.56%	78.76%	70.34%	61.72%	36.87%
TREC			50% Symmetric Noise + 50% Pairflip Noise						50% Random Noise + 50% Pairflip Noise					
	Method	Clean Data	10%	20%	30%	40%	50%	60%	10%	20%	30%	40%	50%	60%
	Co-teaching	88.40%	82.40%	81.20%	77.20%	74.20%	65.40%	47.00%	86.80%	80.20%	76.60%	70.40%	61.00%	36.20%
	Co-teaching+	84.60%	82.80%	81.40%	80.20%	72.40%	70.20%	60.60%	85.00%	78.40%	77.80%	75.40%	68.40%	30.80%
	JoCoR	84.80%	85.00%	81.20%	78.80%	75.00%	67.80%	44.00%	84.80%	80.40%	78.60%	68.60%	62.60%	34.40%
	LSTM$_{DN-H}$	94.20%	91.10%	88.70%	85.30%	83.10%	82.00%	81.00%	**91.80%**	88.50%	86.30%	**84.30%**	**83.60%**	81.00%
	LSTM$_{DN-S}$	**94.40%**	**91.60%**	**88.90%**	**86.70%**	**84.40%**	**82.80%**	**81.30%**	91.40%	**89.70%**	**88.20%**	84.10%	82.80%	**81.10%**
	Peer	78.44%	78.07%	77.65%	75.11%	73.34%	68.92%	45.20%	76.23%	74.88%	73.26%	65.06%	53.73%	28.19%
	CL	82.63%	83.20%	80.76%	77.59%	72.20%	62.31%	48.24%	84.19%	81.47%	74.33%	71.22%	60.09%	33.35%
			Symmetric Noise						Pairflip Noise					
	Method	Clean Data	10%	20%	30%	40%	50%	60%	10%	20%	30%	40%	50%	60%
	Co-teaching	78.43%	76.95%	75.55%	74.86%	72.99%	69.33%	63.54%	77.51%	76.01%	69.59%	65.51%	47.80%	15.64%
	Co-teaching+	76.88%	76.59%	75.84%	75.53%	74.62%	71.82%	67.46%	76.83%	75.46%	74.26%	69.53%	47.08%	15.66%
	JoCoR	77.92%	77.00%	76.17%	74.45%	72.96%	70.12%	62.95%	76.99%	75.12%	70.22%	62.39%	38.14%	15.51%
	LSTM$_{DN-H}$	**93.31%**	91.54%	91.24%	91.01%	88.53%	87.92%	87.66%	91.56%	90.94%	90.55%	88.03%	87.68%	87.45%
	LSTM$_{DN-S}$	**93.31%**	**91.77%**	**91.48%**	**91.07%**	**89.42%**	**88.79%**	**88.52%**	**91.75%**	**91.30%**	**90.87%**	**89.52%**	**88.90%**	**88.37%**
	Peer	74.03%	73.77%	72.68%	72.60%	71.00%	70.59%	65.39%	73.35%	72.16%	71.63%	67.08%	36.11%	17.52%
	CL	80.30%	78.58%	69.97%	63.21%	55.19%	46.86%	37.43%	77.23%	69.32%	61.56%	54.33%	46.12%	36.28%
			White Noise						Uniform Noise					
	Method	Clean Data	10%	20%	30%	40%	50%	60%	10%	20%	30%	40%	50%	60%
	Co-teaching	78.43%	78.46%	78.41%	78.18%	78.05%	77.25%	76.05%	77.07%	76.34%	74.54%	72.61%	63.42%	62.75%
	Co-teaching+	76.88%	77.02%	76.66%	77.07%	76.45%	76.33%	76.86%	76.59 %	76.16%	75.80%	74.11%	71.79%	64.99%
	JoCoR	77.92%	77.88%	77.87%	77.68%	77.39%	76.17%	74.55%	77.09%	75.80%	73.79%	71.42%	67.59%	60.58%
	LSTM$_{DN-H}$	**93.31%**	**93.34%**	**93.25%**	**93.24%**	**93.20%**	92.89%	**93.12%**	91.59%	91.20%	90.83%	90.05%	89.74%	88.51%
	LSTM$_{DN-S}$	**93.31%**	**93.34%**	93.18%	93.11%	93.15%	**93.04%**	93.07%	**91.86%**	**91.44%**	**91.01%**	**90.23%**	**89.88%**	**88.63%**
	Peer	74.03%	74.25%	74.11%	73.68%	73.02%	72.43%	72.16%	73.87%	73.68%	72.10%	71.78%	70.31%	63.80%
	CL	80.30%	80.46%	78.73%	77.75%	74.27%	72.68%	67.35%	78.51%	69.72%	62.82%	54.06%	46.72%	37.18%
Ag-News			50% Symmetric Noise + 50% Pairflip Noise						50% Uniform Noise + 50% Pairflip Noise					
	Method	Clean Data	10%	20%	30%	40%	50%	60%	10%	20%	30%	40%	50%	60%
	Co-teaching	78.43%	76.45%	75.30%	73.63%	71.24%	63.91%	40.82%	78.43%	77.09%	75.94%	73.53%	63.34%	46.24%
	Co-teaching+	76.88%	76.96%	75.79%	74.89%	73.41%	67.49%	38.71%	76.74%	75.99%	74.25%	74.00%	66.46%	43.21%
	JoCoR	77.92%	76.92%	75.17%	74.39%	75.64%	63.03%	75.95%	76.39%	75.64%	72.95%	70.00%	62.72%	47.63%
	LSTM$_{DN-H}$	**93.31%**	91.43%	90.87%	90.16%	87.95%	87.66%	87.12%	91.86%	91.47%	91.09%	90.03%	**89.85%**	**89.37%**
	LSTM$_{DN-S}$	**93.31%**	**91.59%**	**91.25%**	**90.63%**	**89.04%**	**88.20%**	**87.64%**	**92.31%**	**91.85%**	**91.20%**	**90.86%**	89.57%	89.30%
	Peer	74.03%	73.77%	73.50%	72.37%	70.74%	62.59%	46.49%	73.01%	73.16%	72.20%	71.29%	63.63%	44.93%
	CL	80.30%	77.63%	68.42%	60.30%	54.89%	45.21%	36.85%	78.69%	71.37%	64.49%	56.74%	47.55%	39.64%
			Symmetric Noise / Pairflip Noise						White Noise					
	Method	Clean Data	10%	20%	30%	40%	50%	60%	10%	20%	30%	40%	50%	60%
	Co-teaching	72.05%	66.88%	69.98%	64.39%	63.35%	55.28%	38.3%	70.18%	71.22%	68.12%	69.98%	68.94%	67.08%
	Co-teaching+	71.42%	69.98%	69.98%	63.35%	67.49%	52.17%	38.92%	71.84%	66.87%	71.64%	69.98%	69.77%	68.32%
	JoCoR	73.5%	68.94%	70.39%	65.01%	61.49%	**55.90%**	37.06%	69.77%	72.67%	70.6%	70.39%	69.36%	69.15%
	LSTM$_{DN-H}$	59.42%	58.59%	57.26%	56.17%	54.29%	53.46%	**52.23%**	59.42%	59.36%	59.21%	59.14%	59.07%	59.10%
	LSTM$_{DN-S}$	59.62%	59.00%	57.83%	56.33%	64.60%	53.55%	52.13%	59.62%	59.62%	59.61%	59.59%	59.57%	59.53%
	Peer	75.63%	73.23%	71.17%	65.31%	60.69%	42.35%	31.02%	76.11%	76.82%	73.25%	72.07%	71.53%	71.83%
	CL	**88.17%**	**87.55%**	**85.47%**	**74.27%**	**73.03%**	47.51%	28.22%	**88.59%**	**88.38%**	**88.80%**	**86.10%**	**89.00%**	**87.55%**
Chnsenticorp			Symmetric Noise / Pairflip Noise						White Noise					
	Method	Clean Data	10%	20%	30%	40%	50%	60%	10%	20%	30%	40%	50%	60%
	Co-teaching	75.98%	72.05%	71.01%	69.57%	66.05%	48.45%	32.20%	76.81%	75.16%	74.12%	75.36%	75.78%	75.36%
	Co-teaching+	76.19%	73.08%	71.01%	69.77%	64.80%	54.04%	35.40%	74.95%	74.95%	74.21%	76.81%	75.16%	75.57%
	JoCoR	75.78%	76.19%	70.60%	69.15%	63.56%	56.11%	38.92%	78.05%	74.32%	73.71%	75.57%	73.08%	72.26%
	LSTM$_{DN-H}$	62.11%	61.90%	59.61%	57.43%	55.15%	53.40%	**52.10%**	62.11%	62.06%	62.01%	61.98%	61.83%	61.78%
	LSTM$_{DN-S}$	62.31%	62.05%	59.16%	57.42%	54.84%	52.96%	51.89%	62.11%	62.07%	61.90%	61.75%	61.90%	61.85%
	Peer	79.15%	77.83%	74.26%	70.89%	68.35%	**60.32%**	37.11%	79.83%	79.02%	78.83%	78.49%	77.10%	76.60%
	CL	**95.44%**	**91.49%**	**85.06%**	**79.25%**	**70.95%**	50.83%	28.63%	**95.85%**	**94.81%**	**94.19%**	**95.43%**	**93.98%**	**95.02%**
G-Chnsenticorp														

Table 3: The accuracy of seven different models trained on the four datasets with different noise ratios and noise modes respectively

6 Results and Discussion

In our experiments, we evaluate the performance of the models in Table 3 on accuracy which is a widely-used metric in noise learning. We consider five different noise modes (random, symmetric, pairflip, uniform, and white) and some of their combinations on four datasets. We compare the performance of seven widely-used methods and five of them are originally used in the field of computer vision. Hence, we transform them to adapt to the task of text classification. To examine the robustness of the proposed approaches, we set the noise ratio from 0% to 60%. In detail, $p \in \{0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6\}$.

6.1 Effects of Noise Mode

For single mode of noise We find that when white noise is included, the results are comparable with those of the clean dataset. Even if the ratio of noise reaches the maximum, the results are still comparable. We believe that this is because neural networks learn features from text contents. Since the white-noise texts are not related to the original datasets, the results are not greatly affected.

For multi-class classification problems We find that in most cases when the noise ratio increases to 30%, the accuracy of models with pairflip noise is significantly lower than others. We argue that the main reason here is the transfer matrix. When the number of classes exceeds two, the labels with pairflip noise can only transfer to a fixed category or remain unchanged. But for symmetric, uniform, and random transfer matrices, there is a certain possibility of transferring to each category. When the number of classes is two, pairflip transfer matrix is the same as other transfer matrices. Therefore, models involving pairflip noise may have worse accuracy.

For different combinations of noise modes We select symmetric, uniform, pairflip, and random noises to do combinations. We set the total noise ratio from 10% to 60% and assign an equal portion to each noise of the combinations. For most combinations, we find that when combining two noise modes which generally have high classification accuracy, the accuracy of the combination is usually lower than that of the single noise mode. Interestingly, when two noise modes with low accuracy in each single mode are combined, the accuracy may be higher than their single mode's results. For example, in Ag-News we find that choosing the combination of 50% uniform and 50% pairflip noise can achieve better results than their single mode of noise.

6.2 Effects of Noise Rate

For all four datasets, we find that as the noise ratio increases, the performance gradually decreases in most cases. Especially when the noise ratio exceeds 30%, the accuracy drops significantly. However, there are a few exceptions. For instance, including less than 20% of symmetric or white noises would lead to higher accuracy than the clean dataset. We think this may be due to the network's robustness as suggested in (Rolnick et al., 2017) . For symmetric noise, different from other noise transfer matrices, its matrix is an equal division of probabilities except for the diagonal. Then, labels flip with a small equal probability. Due to the fault-tolerance of neural networks, the accuracy can be high. For white noise, when adding a small number of irrelevant texts, the model is consistent and better in its predicted results.

Noise Mode	Noise Ratio %				
	5	10	15	20	25
symmetric	75.97%	73.24%	72.05%	71.52%	71.01%
white	76.03%	76.32%	76.81%	75.23%	75.17%
intrinsic noise	75.83%	70.18%	68.42%	65.97%	63.09%

Table 4: Accuracy of Co-teaching with different noise modes and ratios on G-chnsenticorp

6.3 Effects of Method

For all three categories methods mentioned in Section 5, we find that the data-based method achieves high accuracy in Chnsenticorp and G-Chnsenticorp. The goal of CL is to discover the underlying noise transition pattern which is closer to our noise generation approaches. Therefore, CL can get more accurate results. Compared to Chnsenticorp and G-Chnsenticorp, TREC and Ag-News have more categories than them. This may be the reason for the decreasing results on TREC and Ag-News.

For TREC and Ag-News, the objective-based method, especially LSTM$_{DN-H}$ and LSTM$_{DN-S}$, performs better than other methods. We think that the model can identify the wrong labels by the sample loss value of the training process via modifying the loss function. Hence, based on a suitably constructed loss, it can train a robust deep classifier from the noisy training data and thus can assign correct labels on clean test data.

Among Co-teaching, Co-teaching+, and JoCoR approaches, Co-teaching+ achieves the best result. We argue that this noise learning approach can capture an arbitrary noise function so it can predict a more precise result.

6.4 Effects of Dataset

Compared with Chnsenticorp and G-Chnsenticorp, we find the results on G-Chnsenticorp are significantly better than those on Chnsenticorp. There may be two reasons for this. First, the imbalance of label distribution in Chnsenticorp may affect the results. Second, **intrinsic noise** is an important influencing factor. Interestingly, for results on symmetric or pairflip noise, we find Chnsenticorp has worse results than G-Chnsenticorp at first. But as the noise ratio exceeds 40%, the accuracy on G-Chnsenticorp drops even faster than Chnsenticorp.

Models	Noise Ratio %				
	10	20	30	40	50
Co-teaching$_{FNN}$	83.80%	81.80%	75.80%	70.40%	67.60%
Co-teaching$_{CNN}$	83.34%	80.07%	74.56%	65.44%	62.03%
Co-teaching$_{BERT}$	87.27%	83.77%	77.92%	74.35.67%	70.03%

Table 5: Accuracy of models with different complexities and ratios of random noise on G-chnsenticorp

6.5 Effects of Intrinsic Noise

Because we completely check and relabel the Chnsenticorp dataset, we can analyze the impact of **intrinsic noise** and artificial noise (e.g. symmetric) on the model. We gradually add noise samples to G-chnsenticorp to examine the effects of different **intrinsic noise** ratios on the performance of Co-teaching model.

According to Table 4, we can see that the accuracy with **intrinsic noise** is higher than that of artificial noise. This is because **intrinsic noise** contains more uncertainty than artificial noise which has a definite pattern. There can be a number of reasons for incorrect annotating of data: ambiguity of the correct label (Zhan et al., 2019), annotation speed, human errors, inexperience of annotator, etc. The noise labels generated by these behaviors have no patterns. It is more difficult for the model to capture and counteract these noise labels. Thus, the accuracy of the model with intrinsic noise is negatively affected.

6.6 Effects of Model Complexity

We also examine the influence of model complexity on its performance using different feature extraction layers. We experiment on G-chnsenticorp with random noise. Intuitively, models with different complexities should have different tolerances for noise data. Results in Table 5 also illustrate this: the accuracy of CNN and FNN drops faster than that of BERT as the noise ratio increases. Since the BERT model is pre-trained on a large-scale corpus, it has better generalization ability and can better combat the interference from noise data.

7 Conclusion

In conclusion, we firstly construct a text classification dataset without **intrinsic noise**, then do experiments on these datasets using some sota noise learining methods, and finally draw some useful conclusions about noise learning: **Intrinsic Noise** is more difficult to be learned than other noise modes; many methods do not work well when the noise ratio is higher than 30%; a small amount of white noise can benefit classification methods. This is the first time a benchmark of noise learning for text classification has been established. We construct a dataset without **intrinsic noise** for more accurate evaluations in the noise learning. We present this benchmark to summarize and compare the contributions and weaknesses of previous work in noise learing, to make up for their lack in **intrinsic noise** analysis, and hopefully to provide a reference for future research in the field of noise learning.

In future work, we will build more data without **intrinsic noise** and conduct more in-depth analysis of **intrinsic noise** in other settings. Of course, we will also research a better noise learing method based on the experimental findings of this paper.

Acknowledgements

This work is supported by national key research and development project, No. 2019YFB2102500, National Nature Science Foundations of China under Grant U20B2052.

References

Görkem Algan and Ilkay Ulusoy. 2020. Label noise types and their effects on deep learning. *arXiv preprint arXiv:2003.10471*.

P.S. Sastry Aritra Ghosh, Himanshu Kumar. 2017. Robust loss functions under label noise for deep neural networks. *arXiv preprint arXiv:1712.09482*.

Isaac L. Chuang Curtis G. Northcutt, Lu Jiang. 2019. Confident learning: Estimating uncertainty in dataset labels. *arXiv preprint arXiv:1911.00068*.

Jessy Lin Curtis G. Northcutt, Anish Athalye. 2020. Pervasive label errors in ml benchmark test sets,consequences, and benefits. In *Workshop on Dataset Curation and Security - NeurIPS 2020*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

David Flatow and Daniel Penner. 2017. On the robustness of convnets to training on noisy labels.

Siddhant Garg, Goutham Ramakrishnan, and Varun Thumbe. 2021. Towards robustness to label noise in text classification via noise modeling. *arXiv preprint arXiv:2101.11214*.

Bo Han, Quanming Yao, Tongliang Liu, Gang Niu, Ivor W Tsang, James T Kwok, and Masashi Sugiyama. 2020. A survey of label-noise representation learning: Past, present and future. *arXiv preprint arXiv:2011.04406*.

Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. 2018. Co-teaching: Robust training of deep neural networks with extremely noisy labels. *arXiv preprint arXiv:1804.06872*.

Ryuichiro Hataya and Hideki Nakayama. 2018. Investigating cnns' learning representation under label noise.

Arthur Jacot, Franck Gabriel, and Clément Hongler. 2018. Neural tangent kernel: Convergence and generalization in neural networks. *arXiv preprint arXiv:1806.07572*.

Lu Jiang, Di Huang, Mason Liu, and Weilong Yang. 2020. Beyond synthetic noise: Deep learning on controlled noisy labels. In *International Conference on Machine Learning*, pages 4804–4815. PMLR.

Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. 2018. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *International Conference on Machine Learning*, pages 2304–2313. PMLR.

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

Mingchen Li, Mahdi Soltanolkotabi, and Samet Oymak. 2020. Gradient descent with early stopping is provably robust to label noise for overparameterized neural networks. In *International Conference on Artificial Intelligence and Statistics*, pages 4313–4324. PMLR.

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.

David Rolnick, Andreas Veit, Serge Belongie, and Nir Shavit. 2017. Deep learning is robust to massive label noise. *arXiv preprint arXiv:1705.10694*.

Songbo Tan and Jin Zhang. 2008. An empirical study of sentiment analysis for chinese documents. *Expert Systems with applications*, 34(4):2622–2629.

Brendan Van Rooyen, Aditya Krishna Menon, and Robert C Williamson. 2015. Learning with symmetric label noise: The importance of being unhinged. *arXiv preprint arXiv:1505.07634*.

Andreas Veit, Neil Alldrin, Gal Chechik, Ivan Krasin, Abhinav Gupta, and Serge Belongie. 2017. Learning from noisy large-scale datasets with minimal supervision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 839–847.

Ellen M Voorhees and Dawn M Tice. 1999. The trec8 question answering track evaluation. In *TREC*, volume 1999, page 82.

Fei Wang, Liren Chen, Cheng Li, Shiyao Huang, Yanjie Chen, Chen Qian, and Chen Change Loy. 2018. The devil of face recognition is in the noise. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 765–780.

Hongxin Wei, Lei Feng, Xiangyu Chen, and Bo An. 2020. Combating noisy labels by agreement: A joint training method with co-regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13726–13735.

Yann LeCun Xiang Zhang, Junbo Zhao. 2015. Character-level convolutional networks for text classification. *arXiv preprint arXiv:1509.01626*.

Jun Xu, Hui Li, Zhetong Liang, David Zhang, and Lei Zhang. 2018. Real-world noisy image denoising: A new benchmark. *arXiv preprint arXiv:1804.02603*.

Hongyi Guo Yang Liu. 2019. Peer loss functions: Learning from noisy labels without knowing noise rates. *arXiv preprint arXiv:1910.03231*.

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

Xueying Zhan, Yaowei Wang, Yanghui Rao, and Qing Li. 2019. Learning from multi-annotator data: A noise-aware classification framework. *ACM Transactions on Information Systems (TOIS)*, 37(2):1–28.

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

Aleksandar Zlateski, Ronnachai Jaroensri, Prafull Sharma, and Frédo Durand. 2018. On the importance of label quality for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1479–1487.

Appendices

A Cases of Intrinsic Noise

Table 6 shows some cases of **Intrinsic Noise**.

B Selection of Threshold

During human annotation, we analyze the annotation results of our 10 annotators. Based on the annotation accuracy on the annotators' emotional inclination (the majority agree on one label), we choose 90% and 60% as the annotator agreement thresholds using the 3σ principle of Normal Distribution.

First, we calculate the rate of annotators' agreement on the annotated labels. We find that the accuracy of annotators getting the labels with significantly emotional inclination correct is 92.08%. With regard to those labels without significantly emotional inclination, we assume that the probability of annotating the label positive or negative is 50%. The rate of annotators' agreement on the annotated labels is illustrated as Table 7. The details results are summarized in Table 8.

Based on the annotator agreement rates on emotionally inclined or not inclined samples, we can compute the accuracy of getting those samples correct as follows:

$$\begin{aligned} accuracy(a_i) &= \frac{a_i}{a_i + b_i} \\ accuracy(b_i) &= \frac{b_i}{a_i + b_i} \end{aligned} \tag{1}$$

where a_i and b_i denote the annotator agreement rate on emotionally inclined or not inclined samples respectively.

The accuracy is summarized in Table 9.

We choose the 95% Confidence interval for our results and have the following definitions on intrinsic noise:

As more than 90% of annotations are the same among the annotators, the original sample is significantly emotionally inclined (accuracy > 95%). If the majority annotation is different from the original label, we consider the sample as Fatal noise.

As less than 60% of annotations are the same among the annotators, the original sample is not significantly emotionally inclined (accuracy > 95%). Not matter what the original label is, we consider it as Ambiguous noise.

As 60% to 90% of annotations are the same among the annotators, the original sample is inexact in terms of emotional inclination. If the majority annotation is different from the original label, we consider the sample as Inexact noise.

C Effects of Hyperparameter Settings

To explore other different settings, we conduct experiments using the following uniform setup: Co-teaching model trained the fixed TREC dataset with different ratios of random noise. Here, we explore the effects of learning rates, number of training epochs, and optimizers. Table 10 shows the accuracy on different settings.

For learning rates, we observe that the optimal learning rate increases as the noise ratio increases as expected. We think this is because an appropriately large learning rate can help the model escape from local optimum and increases the model robustness to noise labels. Small learning rates tend to make the model trapped in a locally optimum or overfit the model. It also takes a longer time to train.

For training epochs, we set the same learning rate 1e-3. We find model can gradually fit all data as training epochs increase. But after 50 epochs, the model is over-fit. We have two interesting observations on fitting labels with random mode: a) we do not need to change the learning rate schedule; b) once the fitting starts, it converges quickly.

For optimizers, as the noise ratio increases, the results with RMSprop and Adam optimizers are significantly better than the simple SGD optimizer. The RMSprop and Adam optimizers have the following advantages: first, the gradient of the current batch is used for fine-tuning the final update. Second, the learning rate is adaptive for each parameter. These advantages help to be able to get rid of the local optima.

Case	Source Label	Checked Label	Sample source
What company is being bought by Yahoo and how much is the deal worth ?	HUM	HUM and NUM	TREC
What is the best college in the country ?	HUM	LOC	TREC
Mars water tops science honours.The discovery that salty, acidic water once flowed across the surface of Mars has topped a list of the 10 key scientific advances of 2004.	World	Sci/Tech	Ag-News
Pharma Groups Work on EPC Issues.Sept. 30, 2004 Reacting to calls from pharmaceutical retailers, distributors and manufacturers, EPCglobal has added a new action group to specifically study the pharmaceutical industry.	Sci/Tech	Sci/Tech and Business	Ag-News
比较实惠，旁边有易初莲花，买东西比较方便，还有麦当劳。 (This hotel is not only affordable, but also close to Etsu Lotus, which is convenient for shopping.There's also a McDonald's which is convenient for dining.)	negative	positive	Chnsenticorp
购物较方便，上外滩也近，但房间太小。没有早餐不方便，较为嘈杂，装修较老。 (This hotel is convenient for shopping and close to the Bund. But the disadvantages are small room, no breakfast, noisy environment and old decoration.)	negative	positive and negative	Chnsenticorp
作为酒店的老客户，恐怕以后要做另外的选择了——服务水平在下降，价格却一升再升，再这样下去，下次不会再入住了。(As a regular customer of the hotel, I'm afraid I'll have to make another choice in the future - the service level is declining, but the price is rising. if this continues, I won't stay there again next time.)	positive	negative	Chnsenticorp

Table 6: Examples of **intrinsic noise** in different datasets

Conditions	Rate generation formula
$n\ mod\ 2 = 1$ and $\lceil n/2 \rceil \leq i \leq n$	$rate = C_n^i p^i (1-p)^{n-i} + C_n^{n-i} p^{n-i} (1-p)^i$
$n\ mod\ 2 = 0$ and $i = \lceil n/2 \rceil$	$rate = C_n^i p^i (1-p)^{n-i}$
$n\ mod\ 2 = 0$ and $\lceil n/2 \rceil < i \leq n$	$rate = C_n^i p^i (1-p)^{n-i} + C_n^{n-i} p^{n-i} (1-p)^i$

Table 7: The conditions and the rate generation formulas of annotators' agreement. Here, n is the number of annotators, i denotes the number of annotators who get the same annotation, p denotes the accuracy of annotators getting the labels correct.

Sample type\The rate of same annotation (n=10)	100%	90%	80%	70%	60%	50%
Emotionally inclined	43.82%	37.69%	14.59%	3.35%	0.51%	0.05%
Not Emotionally inclined	0.20%	1.95%	8.79%	23.44%	41.02%	24.61%

Table 8: The rate of annotators agreement on emotionally inclined or not inclined samples. Note that this table only lists annotator agreement rate from 100% to 50% as the dataset is binary. The rates are complementary such as r90% = r10%. All percentages are rounded up.

Accuracy\ The rate of same annotation (n=10)	100%	90%	80%	70%	60%	50%
Emotionally inclined	99.56%	95.07%	62.40%	12.49%	1.22%	0.21%
Not Emotionally inclined	0.44%	4.93%	37.60%	87.51%	98.78%	99.79%

Table 9: The accuracy of getting emotionally inclined or not inclined samples correct.

Hyperparameters		Noise Ratio %				
		10	20	30	40	50
learing rates	1e-2	83.79%	82.10%	75.92%	70.97%	68.60%
	1e-3	83.80%	81.80%	75.80%	70.40%	67.60%
	1e-4	82.56%	81.44%	75.59%	70.11%	66.36%
	1e-5	82.43%	81.32%	75.24%	69.89%	66.12%
	1e-7	82.13%	80.76%	74.68%	69.43%	65.68%
training epochs	10	83.24%	79.74%	73.96%	69.36%	65.07%
	20	83.80%	81.80%	75.80%	70.40%	67.60%
	30	86.55%	81.54%	76.05%	70.94%	67.80%
	50	86.55%	81.66%	76.23%	70.93%	67.81%
	100	86.56%	81.67%	76.23%	71.04%	67.81%
optimizers	SGD	82.43%	78.37%	72.68%	67.60%	65.03%
	Adagrad	82.57%	78.45%	72.97%	67.83%	65.48%
	Momentum	83.64%	79.86%	73.52%	68.25%	66.23%
	RMSprop	85.24%	81.22%	74.26%	69.22%	67.01%
	Adam	83.80%	81.80%	75.80%	70.40%	67.60%

Table 10: Accuracy of Co-teaching model trained fixed TREC dataset with different noise ratios of random noise and hyperparameter settings