Label Noise Robust Curriculum for Deep Paraphrase Identification

Boxin Li*†, Tingwen Liu*†, Bin Wang‡, and Lihong Wang§
*Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
†School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
‡Xiaomi AI Lab, Beijing, China
§National Computer Network Emergency Response Technical Team Coordination Center of China, Beijing, China
{liboxin, liutingwen}@iie.ac.cn, wangbin11@xiaomi.com, wlh@isc.org.cn

***Abstract*—In this paper, we study the effect of label noise on deep learning models for paraphrase identification. Curriculum learning, a learning paradigm that learns easy samples first and then gradually proceeds with hard ones, has shown excellent results in dealing with label noise for deep neural networks (DNNs). However, most previous studies focus on image classification, and design their curriculum only based on training losses of samples, ignoring domain-specific knowledge. In this paper, we propose a predefined curriculum learning based framework, incorporating both training losses of samples and domain-specific knowledge, to train robust deep models for paraphrase identification (PI) with label noise. Through extensive experiments on two popular PI benchmarks, we show that 1)the performance of the deep paraphrase identification model can drop sharply at the case of severe label noise; 2)our approach can significantly improve generalization performance of deep networks trained on corrupted data especially at extremely high levels of label noise; 3)our method can outperform several state-of-the-art label corruption robust methods.**

***Index Terms*—paraphrase identification, label noise, curriculum learning, deep learning**

I. Introduction

Paraphrase Identification (PI), detecting whether two sentences convey the same meaning, has been widely applied in community-based question answering [1] and conversational assistant systems [2]. Nowadays, a large number of end-to-end deep neural networks (DNNs) for PI have been proposed and achieved promising results [3–7]. However, the presence of label noise can seriously impair the performance of DNNs, which has been extensively studied in image classification [8–14], but few works exist in the field of PI. While label noise widely exists in the existing datasets of paraphrase identification, due to automatic labeling [15, 16] and non-expert labeling [17]. To fill this gap, we study the effect of label noise on deep learning models for PI, and how to advance them at the case of label noise.

Few methods have been proposed to leverage curriculum learning [18] to train robust deep models, which are inspired by studies from Arpit et al. [19] that DNNs tend to learn easy samples first, and then gradually adat to hard ones. But most of them come from the image domain. For example, Jiang el al. [9] proposed a data-driven curriculum (called *MentorNet*) for image classification, which needs an additional network and some clean labeled samples to output a dynamic curriculum. Zhu et al. [20] proposed a self-paced curriculum framework for face recognition. Existing studies have shown that this direction is promising. However, they are mainly based on losses of training samples, ignoring domain-specific knowledge.„

We argue that an ideal curriculum needs to incorporate domain-specific prior knowledge. To this end, this paper proposes a predefined curriculum learning based framework, combining both training losses and prior knowledge, to train robust deep models for PI with label noise, which consists of two stages: curriculum design and model training.

In the first stage (*curriculum design*), we employ a hybrid measurement, composed of a linear weighting of two practical measures, to automatically measure the noise probability (complexity) for each paraphrase pair. Then, we sort all training samples in ascending order according to the final hybrid measurement to generate our curriculum.

In the second stage (*model training*), we divide all training samples equally into several groups, according to the above generated curriculum. As such, we firstly utilize simple and low-level noisy instances from the preceding groups to train a basic model, and then gradually augment hard and high-level noisy samples. In this way, deep models can expectantly be affected by noisy data as late as possible, and meanwhile take full advantage of clean data.

To verify the effectiveness of our approach, extensive experiments are conducted on two popular English and Chinese PI benchmarks, QQP [6] and LCQWC [21]. Experimental results show that our method can significantly improve generalization performance of deep networks trained with corrupted labels, and can outperform other state-of-the-art methods.

The main contributions of this paper are as follows:

(1) To the best of our knowledge, we are the first to study the effect of label noise on deep learning models for paraphrase identification.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

(2) We propose a newly curriculum learning based framework to train robust deep models for paraphrase identification with label noise. And we develop two measures to define the noise probability (complexity) of paraphrase sentence pairs, and find that the linear weighted combination of them can achieve better results.

(3) We conduct extensive experiments on both Chinse and English PI benchmarks to verify the effectiveness and robustness of our method.

II. Related Work

A. Paraphrase Identification

The goal of paraphrase identification(PI) is to judge whether two sentences express the same meaning. Nowadays, a large number of deep PI models have been proposed and achieved promising results [3–7]. However, none of them were designed explicitly to tackle the issue of label noise. To make up for this deficiency, this paper explores how to improve the state-of-the-art deep models for PI in the setting of corrupted labels.

B. Curriculum Learning

Curriculum learning, inspired by the learning principle from humans and amimals, is a learning paradigm that trains easy samples first, and then gradually proceeds with hard ones, proposed by Bengio et al. [18]. After that, Kumar et al. [22] presented a dynamic curriculum called self-paced learning, which prefers samples of small loss. Following were a series of improved versions of self-paced learning [23–25]. Moreover, some previous studies in the literature showed the usefulness of curriculum learning in dealing with noisy samples [26–28]. Unlike these works, our approach targets paraphrase identification, and leverages a static predefined curriculum, which combines both training losses of instances and domain-specific knowledge.

C. Learning from noisy labels

Learning from noisy labels is a long-lasting topic in the machine learning community, and can date back to three decades ago [10]. Here, we only review approches related to our work, aiming to address the issue of label noise for DNNs.

These label noise robust methods can roughly be divided into the following three categories.

i) *Developing robust loss functions.* The goal of these methods is to formulate noise-robust loss functions for DNNs. For example, MAE (Mean Absolute Error) [29] is proposed to resist label noise for DNNs, but can increase difficulty in training. To address this issue, Zhang et al. [30] presented the GCE (Generalized Cross Entropy) loss function, an extention of MAE and CCE (Categorical Cross Entropy). The advantage of these methods is their generality and non-invasive to the existing architectures and algorithms.

ii) *Modifying the outputs of base models.* This line of research mainly focus on estimating the noise transition matrix. For instance, Sukhbaatar et al. [11] estimated the noise transiton matrix by introducing an extra noise layer on top of the base network, to match the label noise distribution. However, the noise transition matrix is nontrival to learn. To address this issue, Goldberger et al. [12] put on an additional softmax layer with the base model. Hendrycks et al. [14] made use of a small fraction of trusted data to estimate the noise matrix. Jindal et al. [31] applied appropriate initialization to the noise layer, introduced on top of the target model, and achieved impressive results for text classification. Although many attempts have been conducted, it is still not easy to accurately estimate the noise transition matrix.

iii) *Using selected samples.* To get rid of estimation of the noise transition matrix, another branch of methods focus on training on selected samples [9, 20, 32, 33]. The core problem of these methods is the identification of noisy samples, of which the representative ones are MentorNet [9], SPDL [20], Co-teaching [33]. The MentorNet pretrains an extra noise classifier with some clean annotated data, and provides a learned curriculum for the target network. Another version of the MentorNet leverages the self-paced curriculum, at the case of no access to clean data. As an extention, the SPDL [20] pre-trains the target network on some clean data before using the self-paced curriculum framework. The Co-teaching [33] trains two networks simutaneously, and updates models only utilizing a small fraction of small-loss samples provided by the peer network.

However, most of the above methods target image classification. In this paper, we present a curriculum learning based framework to train robust DNNs for paraphrase identification with corrupted labels. Our work is very similar to the MentorNet [9] and SPDL [20], but requires no clean data. Moreover, to encode richer information to recognize noisy samples, our method predefines a static curriculum, which can not only leverage training loss information, but also utilize prior knowledge excavated from the target task.

III. Method

A. Label Noise

Paraphrase identification is commonly tackled as a supervised binary classification problem. Most previous studies focus on learning deep models with good generalization performance on the large-scale clean labeled training dataset $D = \{(X_1, y_1), (X_2, y_2), ..., (X_n, y_n)\}$, where X_i is the *ith* paraphrase sentence pair, and $y_i \in \{0, 1\}$ is the golden label. However, such large-scale refined datasets are usually expensive and time-consuming to collect.

Thus, this paper instead focuses on training robust deep models for papraphrase identification with corrupted labels, generated artificially by injecting noise into clean data. Here, we take class-conditioned *uniform label flip noise.* Specifically, given noise rate p ($0<p<1$, also known as *noise level*), we can firstly produce a corruption transition matrix:

$$C = pI^K + \frac{(1-p)}{(K-1)}(II^K - I^K), \tag{1}$$

where K denotes the category number, I^K represents the identity matrix, and II^K is the matrix with all ones. $C_{i,j} = p\{\tilde{y} = j|y = i\}$ means the probability of a sample from category i is wrongly labeled as category j, where $\tilde{y}$ is noisy label and y is clean label. After flipping the labels of training samples drawn according to the transition matrix C, we can get a noisy training dataset with a pre-given noise rate p. The validation set uses the same method of adding noise. We do not alter the test dataset.

B. Label Noise Robust Curriculum

It is easy to think of utilizing training losses to measure the noise probability (complexity) of a sample. As such, we firstly train a deep model on the corrupted training dataset for several epochs, and record the training losses of each sample in each epoch. The key problem is how to use these losses to generate some appropriate metric so as to distinct noisy samples significantly. Studies by Arpit et al. show that DNNs fit fastly with clean samples and slowly with noisy samples. Based on these findings, we design an easy and effective metric, called *loss based noise metric*(*NM_loss*), to compute the noise probability (complexity) of a sample. We define *NM_loss* as the mean value of a sequence of losses for a training sample in the first k epochs as follows:

$$NM\_loss(X_i) = \frac{1}{k}\sum_{j=1}^{k} loss_j(X_i), \tag{2}$$

where $X_i = (s_{i1}, s_{i2})$, s_{ij} represents the jth sentence in the sample X_i, $j \in \{1, 2\}$ and k is a parameter.

In addition to the above metric based on training losses, we further develop another metric dedicated to the paraphrase identification task. This metric steps from a common phenomenon: ***paraphrase pairs with small literal similarity tends to be hard samples, and non-paraphrase pairs with large literal similarity tends to be hard samples, and vice versa***. We implement this idea with *Jaccard similarity coefficient* [1] between two sentences. In this way, *similarity based noise metric* (*NM_sim*) of a given sample X_i is defined as follows:

$$NM\_sim(X_i) = \begin{cases} 1 - Jac(s_{i1}, s_{i2}), X_i \; is \; paraphrase, \\ Jac(s_{i1}, s_{i2}), X_i \; is \; non-paraphrase, \end{cases} \tag{3}$$

where *Jac* is the abbreviation of *Jaccard*.

Based on the above two measures, we get the final noise metric for a given sample X_i as follows:

$$NM(X_i) = \lambda NM\_loss(X_i) + (1-\lambda)NM\_sim(X_i), \tag{4}$$

where λ is a tradeoff parameter.

C. Label Noise Robust Curriculum Framework to Learn Deep Models

Algorithm 1 Predefined curriculum learning for deep paraphrase identification

```
Input: Corrupted training dataset $\mathcal{D}_t$; Corrupted Validation dataset $\mathcal{D}_v$; Maximum epochs $T$; Learning rate $\alpha$; Mini-batch size $H$; A deep paraphrase identification model $f(\theta)$; tradeoff parameter $\lambda$; Sample group number $N_g$; Epoch number $k$ to compue loss-based noise metric.
Output: Optimal parameter $\theta^*$ for $\mathcal{D}_t$.
Train the classifier on $\mathcal{D}_t$ to get and store training losses of each sample in each epoch.
Calculate the noise probability for each sample by metrics introduced in (4).
Sort training samples according to noise probability in ascending order to get the sorted dataset $\widetilde{\mathcal{D}_t}$.
Initialize the network $f(\theta)$.
$unit\_num \leftarrow size(\widetilde{\mathcal{D}_t})/N_g$
for $j = 1 \rightarrow N_g$ do
    $sub\_dataset = \widetilde{\mathcal{D}_t}[0 : unit\_num * j]$
    for $t = 1 \rightarrow T$ do
        Fetch a mini-batch $\{(x_i, y_i)\}_1^H$ from $sub\_dataset$ randomly.
        Update the network $f(\theta)$ with SGD.
    end for
    Evaluate $f(\theta)$ on $\mathcal{D}_v$.
end for
return the best model $f(\theta^*)$ evaluated on $\mathcal{D}_v$.
```

Our proposed predefined curriculum learning framework is presented in Algorithm 1. First of all, we prepare the curriculum for the model training stage. To get the loss-based metric, we train a deep classifier network $f(\theta)$ on the corrupted dataset $\mathcal{D}_t$ for T epochs, and get the loss sequence of each sample. Similarity-based metric is generated during the data preparation phase. Applying Formula (4), we can get the final noise metric for each sample. Next, based on the final noise metric of each sample, we sort $\mathcal{D}_t$ in ascending order to get a sorted dataset $\widetilde{\mathcal{D}_t}$. For $\widetilde{\mathcal{D}_t}$, we divide it into N_g groups equally. Our method is based on the curriculum learning framework. That is to say, we retrain the deep classifier network $f(\theta)$ by feeding simple and low-level noisy samples first, and gradually adding hard and high-level noisy samples. In this way, our proposed framework can not only get the benefits from clean samples, but also effectively avoid the damage to the generalization performance caused by noisy samples.

D. Classifier

Note that our approach is a general framework to address the issue of label noise for paraphrase identification

[1] https://en.wikipedia.org/wiki/Jaccard_index

(PI), which can adapt to any implementation of deep text matching models. In this paper, to tradeoff the balance between efficiency and effectiveness, we use RE2 [5] as our base model, as RE2 can achieve promising performace comparing to the state-of-the-art with remarkable efficiency for text matching tasks.

The overall architecture of RE2 consists of five layers: the embedding layer, the alignment layer, the fusion layer, the pooling layer and the prediction layer. To save space, we only briefly introduce them here. We introduce readers to see more details in [5].

Embedding Layer. For two sentences s^1 and s^2, where $s^1 = (s^1_1, s^1_2, ..., s^1_{l_1})$, $s^2 = (s^2_1, s^2_2, ..., s^2_{l_2})$ with l_1 and l_2 tokens respectively, the goal of the embedding layer is to transform tokens in each sentence into fixed length word embeddings in order to get better representations.

Alignment Layer. The purpose of this layer is to compute the aligned representations by the attention mechanism:

$$e_{ij} = F(s^1_i)^{\mathrm{T}} F(s^2_j), \tag{5}$$

where $F(\cdot)$ is a single-layer feed-forward network. After calculating the similarity score e_{ij} between s^1_i and s^2_j, the output of this layer is as follows:

$$\begin{aligned} a^1_i &= \sum_{j=1}^{l_2} \frac{exp(e_{ij})}{\sum_{k=1}^{l_2} e_{ik}} s^2_j, \\ a^2_j &= \sum_{i=1}^{l_1} \frac{exp(e_{ij})}{\sum_{k=1}^{l_1} e_{kj}} s^1_i, \end{aligned} \tag{6}$$

Fusion Layer. To better get local interactive information between two sentences, RE2 provides three perspectives in the fusion layer, and takes the concatenation of their results as the final output of this layer.

$$\begin{aligned} f^{1,1}_i &= G_1([s^1_i; a^1_i]), \\ f^{1,2}_i &= G_2([s^1_i; s^1_i - a^1_i]), \\ f^{1,3}_i &= G_3([s^1_i; s^1_i \circ a^1_i]), \\ f^1_i &= G([f^{1,1}_i; f^{1,2}_i; f^{1,3}_i]) \end{aligned} \tag{7}$$

where G_1, G_2, G_3 and G are one-layer feed-forward networks with independent parameters, $[;]$ denotes the concatenation operation, and $\circ$ represents element-wise multiplication.

Pooling Layer. Through the fusion layer, every token in each sequence corresponds to a rich contextual representation. The goal of the pooling layer is to covert each sequence into a fixed-length vector v_1 (or v_2) using pooling operations, such as max pooling.

Prediction Layer. Like most previous works, the prediction layer takes two vector representention v_1 and v_2 from the pooling layer as inputs, and outputs the prediction probability:

$$\hat{y} = H([v_1; v_2; |v_1 - v_2|; v_1 \circ v_2]), \tag{8}$$

where H is a multi-layer perceptron with an additional softmax layer.

What's mentioned above is only a simple version of RE2, where the alignment layer and the fusion layer together can be viewed as a block. Moreover, to produce richer features, RE2 stacks several blocks with the help of residual connections. In detail, the input of the n-th block $b^{(n)} (n \geq 2)$ is the concatenation of the output of the first block $b^{(1)}$ (the output of the embedding layer) and the summation of the outputs from the previous two blocks:

$$b^{(n)} = [b^{(1)}; o^{(n-1)} + o^{(n-2)}], \tag{9}$$

In this setting, the input of the pooling layer comes from the output of the last block.

IV. Experiment

In this section, to empirically demonstrate the effectiveness of our method, on two popular datasets from different languages (Chinese and English), we evaluate the performance of our proposed framework and compare our results with other state-of-the-art methods.

A. Datasets

Chinese paraphrase dataset. LCQMC [21] is a newly public large-scale Chinese paraphrase dataset. It has been already splited into three parts: a training set containing 238,766 instances, a development set with 8,802 instances, and a test set with 12,500 instances. We find that there are some completely duplicate instances in the training set, and even some instances with opposite labels while sharing the same sentence pair. In order to obtain a relatively clean training set, we remove the instances with opposite labels, and keep only one copy for those completely duplicate instances. In the end, we get 238,055 training instances. The Jaccard similarity between two sentences is calculated with Chinese phrases, segmented by *jieba*[2], a common Chinese text segmentation tool.

English paraphrase dataset. QQP [6] is a large-scale English paraphrase dataset and also consists of various question pairs. It contains 404,279 question pairs with binary labels to tell whether they are duplicate or not. We randomly split it into three parts: a training set with 384,279 instances, a validation set with 10,000 instances and a test set with 10,000 instances. Moreover, words are lowercased and tokenized with the nltk toolkit [34] before calculating sentence similarity.

Noise dataset generation. For each task, we construct noisy datasets by switching the labels of randomly selected clean training data and validating data with a noise rate parameter, ranging in $[0.1, 0.2, 0.3, 0.4]$.

[2] https://github.com/fxsjy/jieba

B. Baselines

In addition to the base deep model (*Standard*) trained directly on noisy datasets, we compare our method with some other state-of-the-art baselines. Note that most of them are tailored for image classification, hence we reimplement them on top of our base model, and choose the appropriate hyperparameters on the validation set.

1) **Standard**: A simple and effective deep text matching model, taken as our base deep model, proposed by Yange et al. [5].

2) **Co-teaching**: A method proposed by Han et al. [33], which trains two networks simutaneously and updates models using only some fraction of small-loss training instances from the peer network.

3) **Forward**: A classic label noise robust approach [13] by estimating the noise transition matrix. Here we utilize the true noise transition matrix known in advance to provide a stronger baseline.

4) **Tq**: A label noise robust function, proposed by Zhang et al. [30], as an extention of MAE (Mean Absolute Error) and CCE (Categorical Cross Function) loss functions.

5) **NMwRegu01**: A newly proposed method [31] by introducing a noise layer on top of the target model and applying appropriate initialization methods, to achieve impressive results for text classification. We apply the optimal initialization strategy following their work.

6) **Self-paced**: A curriculum learning based method with a dynamic curriculum [25], provided only according to training losses of each sample.

C. Implementation Details

Note that the goal of this work is to advance the performance of the deep base model at the case of corrupted labels. To this end and for a fair comparison, the same base deep model *RE2* [3] and hyperparameters are utilized for all compared approaches.

All methods are implemented with PyTorch and trained on Tesla K80 GPUs. The parameter settings and implementation details are listed as follows. We train all models for 10 epochs. In all experiments, we take He initialization [35] to initialize model parameters and normalize them by weight normalization [36]. In order to improve the base model's performance, we also initialize the word embedding layer with the pretrained word embeddings and fix them during training. For the Chinese corpus, we use the publicly available *word2vec* Chinese word vectors with 300 dimensions trained on Sogou news [37]. For the English corpus, we utilize the *GloVe* embeddings [38] with 300 dimensions. We take GeLU as our activation function. The batch size is 100. The max sequence length is 64. For the base model RE2, we set the hidden size to 150 and the kernel size to 3, and apply dropout with a probability of 0.2 before every convolutional layer and fully-connected layer. The numbers of blocks and convolutional encoders are set to 1 and 2 respectively for efficient training. We train all models using the Adam optimizer with the default parameters β_1 and β_2 to be 0.9 and 0.999 respectively and minimize the same cross entropy loss. The initial learning rate is set to 0.001. We apply gradient clipping of 10.0. Accurancy is used as the evaluation metric for all experiments.

D. Results and Discussion

In this section, we compare our framework with other state-of-the-art methods on two popular paraphrase identification datasets. Table 1 reports the testing classificattion accurancy on both datasets in the presence of label noise artificially generated at random. The results from *Standard* method show that the performance of the deep paraphrase identification model can drop sharply at the case of severe label noise. For example, when the noise level is 40%, the accuracy of the base model drops 11.89% and 7.98% on QQP and LCQMC respectively, compared to the results on clean data. It can also be observed that our method outperforms significantly the *Standard* method on two different language corpus at all level of label noise, which shows the effectiveness and good generalization of our method. Moreover, in most cases, our approach performs better than other state-of-the-art methods. Especially the advantage over the Self-paced method demonstrates the usefulness of incorporating prior knowledge from the target task into curriculum learning process. More interestingly, we also observe an improvement of 0.6% and 2.53% on QQP and LCQMC respectively over the baselines with clean labels, perhaps due to label noise inherent in the datasets.

Furthermore, to demonstate the robustness of our method, we also conduct various experiments to verify: 1) the effect of different data size; 2) the effect of different noise metric balance factor λ; 3) the effect of different group number N_g; and 4) the effect of the different first k epoch.

1) *Effect of Different Data Size.* To verify the generalization ability of label noise robust methods on datasets with different sizes, we conduct experiments by varying the training data size of QQP. Specifically, we randomly select 10,000, 20,000, 50,000 and 100,000 instances as training set respectively, and retain the original validation and test set. We apply the same strategy to inject label noise. The results are shown in Table 2. As expected, on different size of datasets, our method outperforms significantly the *Standard* baseline, and performs better than other state-of-the-art methods.

Due to the limit of space, we only investigate the last three factors on QQP dataset with 10,000 training instances and 10% random noise.

2) *Effect of Different Noise Metric Balance Factor*

Fig. 1 shows the effect of different noise metric balance factor on classification performance. We vary the noise metric balance factor on x-axis by fixing the group number

[3] https://github.com/alibaba-edu/simple-effective-text-matching

Table1: Test performance for entire datasets QQP and LCQMC

Database	QQP					LCQMC				
Noise Rate (%) / Method	0	10	20	30	40	0	10	20	30	40
Standard [5]	82.87	81.38	80.94	76.31	70.98	81.45	79.74	80.11	76.58	73.47
Tq [30]	\	79.78	78.62	76.18	74.96	\	78.62	77.30	73.74	75.13
NMwRegu01 [31]	79.45	82.64	81.01	76.02	71.93	82.14	79.29	79.84	73.16	72.49
Forward [13]	\	81.66	80.22	77.47	71.54	\	81.15	80.82	76.91	74.34
Co-teaching [33]	\	81.56	78.37	75.74	71.22	\	81.11	79.98	78.18	75.86
Self-paced [25]	81.45	82.01	80.23	77.70	71.99	82.99	82.14	82.28	78.44	**76.71**
Ours	**83.47**	**82.69**	**81.55**	**78.77**	**74.99**	**83.98**	**82.75**	**82.91**	**78.78**	76.26

Table2: Effect of data size on label noise classification performance for QQP

Database	QQP_10000					QQP_20000				
Noise Rate (%) / Method	0	10	20	30	40	0	10	20	30	40
Standard [5]	73.25	72.67	71.04	65.72	61.20	75.23	73.53	72.39	69.24	62.14
Tq [30]	\	72.07	70.33	66.93	62.30	\	73.19	71.27	69.67	62.49
NMwRegu01 [31]	71.58	72.18	71.30	64.95	62.19	74.56	73.14	71.78	69.32	62.24
Forward [13]	\	71.73	70.81	65.55	61.66	\	73.08	72.55	70.24	64.78
Co-teaching [33]	\	71.80	68.35	61.13	58.73	\	73.51	72.21	68.36	62.44
Self-paced [25]	72.10	71.01	69.91	65.33	60.90	74.96	**74.43**	71.28	69.98	64.29
Ours	**73.91**	**73.92**	**71.71**	**68.61**	**65.88**	**76.85**	74.14	**72.91**	**70.63**	**67.65**
Database	QQP_50000					QQP_100000				
Noise Rate (%) / Method	0	10	20	30	40	0	10	20	30	40
Standard [5]	77.52	76.16	74.43	72.18	64.99	78.68	77.91	76.53	74.72	69.78
Tq [30]	\	76.37	74.72	70.99	67.83	\	77.66	76.79	73.32	70.71
NMwRegu01 [31]	74.47	75.50	75.54	70.30	64.78	74.91	77.72	76.73	73.96	68.15
Forward [13]	\	76.24	74.81	72.33	66.69	\	78.11	77.15	74.60	70.59
Co-teaching [33]	\	75.07	73.71	72.22	62.54	\	77.82	74.20	72.07	69.02
Self-paced [25]	77.46	76.08	74.49	72.57	64.56	79.09	78.11	77.11	73.71	68.31
Ours	**78.82**	**78.52**	**76.69**	**72.62**	**70.41**	**80.47**	**78.46**	**77.18**	**75.45**	**71.37**

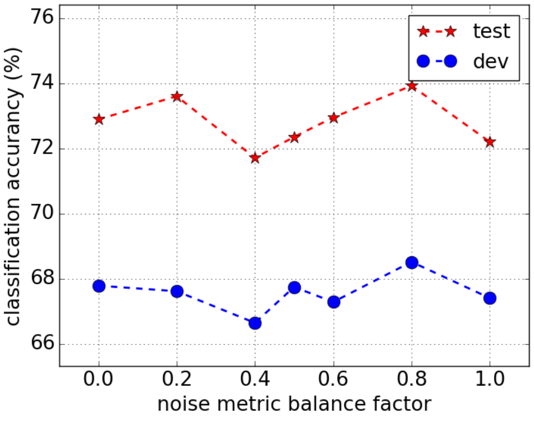

Fig. 1: Effect of noise metric balance factor λ with k=6 and $N_g = 3$.

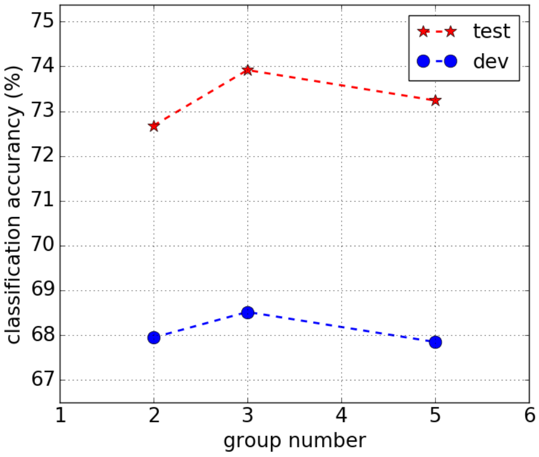

Fig. 2: Effect of different group number N_g with k=6 and $\lambda = 0.8$.

N_g to 3 and the epoch number k to 6. In Fig. 1, we find the optimal value of the balance factor λ is 0.8, between 0.0 and 1.0, which demonstrats the effectiveness of our motivation that the combination of the two proposed noise metrics can achieve better results.

3) *Effect of Different Group Number*

We also observe the effect of different group number on classification performance. In this setting, we vary the group number on x-axis by fixing the noise metric balance factor λ to 0.8 and the epoch number k to 6. In Fig. 2, we can find that the optimal value of N_g is 3. Note that it will degenerate to the *Standard* baseline when N_g is set to 1. Meanwhile, raising N_g would increase computational cost. Hence, we make a tradeoff and set N_g to 3 for all experiments, the suitability of which has also been demonstrated by empirical results here.

4) *Effect of Different First k Epoch*

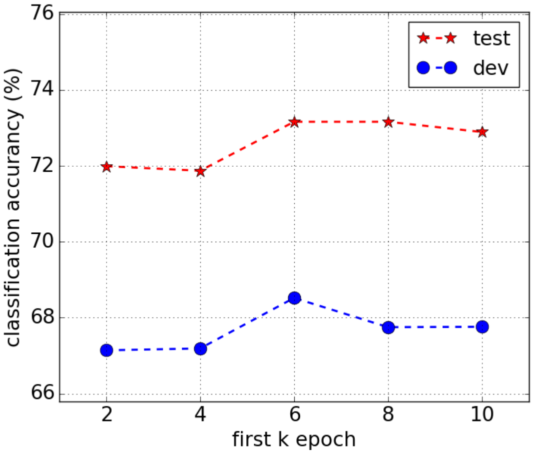

Fig. 3: Effect of different first k epoch with $N_g = 3$ and $\lambda = 0.8$.

We further investigate the effect of different first k epoch. In Fig. 3, we plot the classification performance against epoch k on x-axis by fixing group number N_g to 3 and noise metric balance factor λ to 0.8. As shown in Fig. 3, the optimal value of k is around 6. And a small k (2) and an overly large k (10) often lead to poor performance. A probable explanation is that when k is small, the model training is not enough to significantly distinguish between clean and noisy samples; and when k becomes large, the model starts to remember noisy samples, as a result, the losses of noisy samples drop sharply, and meanwhile the losses of clean samples can be turbulent. Hence, we set k to 6 for all of our experiments, as it usually achieves better results on the validatation set.

Conclusions and Future Work

In this paper, we study the effect of label noise on deep learning models for paraphrase identification, and propose a robust curriculum learning based framework to advance the performance of the deep base model at the case of corrupted labels. We show that the performance of the deep paraphrase identication model can drop sharply at the case of severe label noise. Extensive experiments on two public English and Chinese paraphrase identification benchmarks show that our approach can significantly improve generalization performance of deep networks trained on corrupted data especially at extremely high levels of label noise, and can meanwhile outperform several state-of-the-art label corruption robust methods.

In the future, we will extend our method to more challenging real-world noisy paraphrase identification datasets, such as those datasets collected by automatic paraphrase generation or predicted by existing classifiers. Our method may also adapt to weakly-supervised domain adaptation for paraphrase identification with label noise present in the source domain.

Acknowledgment

This work was supported by the Strategic Priority Research Program of the Chinese Academy of Sciences (No.XDC02040400). We would like to thank anonymous reviewers for the insightful comments.

References

[1] D. Liang, F. Zhang, W. Zhang, Q. Zhang, J. Fu, M. Peng, T. Gui, and X. Huang, “Adaptive multi-attention network incorporating answer information for duplicate question detection,” 2019.

[2] F.-L. Li, M. Qiu, H. Chen, X. Wang, X. Gao, J. Huang, J. Ren, Z. Zhao, W. Zhao, L. Wang *et al.*, “Alime assist: An intelligent assistant for creating an innovative e-commerce experience,” in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management.* ACM, 2017, pp. 2495–2498.

[3] Q. Chen, X. Zhu, Z. Ling, S. Wei, H. Jiang, and D. Inkpen, “Enhanced lstm for natural language inference,” *arXiv preprint arXiv:1609.06038*, 2016.

[4] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” *arXiv preprint arXiv:1810.04805*, 2018.

[5] R. Yang, J. Zhang, X. Gao, F. Ji, and H. Chen, “Simple and effective text matching with richer alignment features,” *arXiv preprint arXiv:1908.00300*, 2019.

[6] Z. Wang, W. Hamza, and R. Florian, “Bilateral multi-perspective matching for natural language sentences,” *arXiv preprint arXiv:1702.03814*, 2017.

[7] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, “Improving language understanding by generative pre-training. 2018,” 2016.

[8] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, “Understanding deep learning requires rethinking generalization,” *arXiv preprint arXiv:1611.03530*, 2016.

[9] L. Jiang, Z. Zhou, T. Leung, L.-J. Li, and L. Fei-Fei, “Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels,” *arXiv preprint arXiv:1712.05055*, 2017.

[10] D. Angluin and P. Laird, “Learning from noisy examples,” *Machine Learning*, vol. 2, no. 4, pp. 343–370, 1988.

[11] S. Sukhbaatar, J. Bruna, M. Paluri, L. Bourdev, and R. Fergus, “Training convolutional networks with noisy labels,” *arXiv preprint arXiv:1406.2080*, 2014.

[12] J. Goldberger and E. Ben-Reuven, “Training deep neural-networks using a noise adaptation layer,” in *ICLR 2017 : International Conference on Learning Representations 2017*, 2017.

[13] G. Patrini, A. Rozza, A. K. Menon, R. Nock, and L. Qu, “Making deep neural networks robust to label noise: A loss correction approach,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2233–2241.

[14] D. Hendrycks, M. Mazeika, D. Wilson, and K. Gimpel, “Using trusted data to train deep networks on labels corrupted by severe noise,” in *NIPS 2018: The 32nd Annual Conference on Neural Information Processing Systems*, 2018, pp. 10 456–10 465.

[15] G. S. Tomar, T. Duque, O. Täckström, J. Uszkoreit, and D. Das, “Neural paraphrase identification of questions with noisy pretraining,” *arXiv preprint arXiv:1704.04565*, 2017.

[16] A. Uva, D. Bonadiman, and A. Moschitti, “Injecting relational structural representation in neural networks for question similarity,” *arXiv preprint arXiv:1806.08009*, 2018.

[17] W. Lan, S. Qiu, H. He, and W. Xu, “A continuously growing dataset of sentential paraphrases,” in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017, pp. 1224–1234.

[18] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, “Curriculum learning,” in *Proceedings of the 26th Annual International Conference on Machine Learning*, 2009, pp. 41–48.

[19] D. Arpit, S. JastrzÄŹbski, N. Ballas, D. Krueger, E. Bengio, M. S. Kanwal, T. Maharaj, A. Fischer, A. Courville, Y. Bengio, and S. Lacoste-Julien, “A closer look at memorization in deep networks,” in *ICML’17 Proceedings of the 34th International Conference on Machine Learning - Volume 70*, vol. 70, 2017, pp. 233–242.

[20] P. Zhu, W. Ma, and Q. Hu, “Self-paced robust deep face recognition with label noise,” in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 2019, pp. 425–435.

[21] X. Liu, Q. Chen, C. Deng, H. Zeng, J. Chen, D. Li, and B. Tang, “Lcqmc:a large-scale chinese question matching corpus,” in *COLING 2018: 27th International Conference on Computational Linguistics*, 2018, pp. 1952–1962.

[22] M. P. Kumar, B. Packer, and D. Koller, “Self-paced learning for latent variable models,” in *Advances in Neural Information Processing Systems 23*, 2010, pp. 1189–1197.

[23] J. S. Supancic and D. Ramanan, “Self-paced learning for long-term tracking,” in *2013 IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 2379–2386.

[24] L. Jiang, D. Meng, S.-I. Yu, Z. Lan, S. Shan, and A. Hauptmann, “Self-paced learning with diversity,” in *Advances in Neural Information Processing Systems*, 2014, pp. 2078–2086.

[25] L. Jiang, D. Meng, Q. Zhao, S. Shan, and A. G. Hauptmann, “Self-paced curriculum learning,” in *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[26] E. Sangineto, M. Nabi, D. Culibrk, and N. Sebe, “Self paced deep learning for weakly supervised object detection,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 3, pp. 712–725, 2018.

[27] X. Chen and A. Gupta, “Webly supervised learning of convolutional networks,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1431–1439.

[28] C. Liu, S. He, K. Liu, and J. Zhao, “Curriculum learning for natural answer generation.” in *IJCAI*, 2018, pp. 4223–4229.

[29] A. Ghosh, H. Kumar, and P. Sastry, “Robust loss functions under label noise for deep neural networks,” in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[30] Z. Zhang and M. Sabuncu, “Generalized cross entropy loss for training deep neural networks with noisy labels,” in *Advances in neural information processing systems*, 2018, pp. 8778–8788.

[31] I. Jindal, D. Pressel, B. Lester, and M. Nokleby, “An effective label noise model for dnn text classification,” *arXiv preprint arXiv:1903.07507*, 2019.

[32] E. Malach and S. Shalev-Shwartz, “Decoupling” when to update” from” how to update”,” in *Advances in Neural Information Processing Systems*, 2017, pp. 960–970.

[33] B. Han, Q. Yao, X. Yu, G. Niu, M. Xu, W. Hu, I. Tsang, and M. Sugiyama, “Co-teaching: Robust training of deep neural networks with extremely noisy labels,” in *Advances in neural information processing systems*, 2018, pp. 8527–8537.

[34] E. Loper and S. Bird, “Nltk: the natural language toolkit,” *arXiv preprint cs/0205028*, 2002.

[35] K. He, X. Zhang, S. Ren, and J. Sun, “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification,” in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.

[36] T. Salimans and D. P. Kingma, “Weight normalization: A simple reparameterization to accelerate training of deep neural networks,” in *Advances in Neural Information Processing Systems*, 2016, pp. 901–909.

[37] S. Li, Z. Zhao, R. Hu, W. Li, T. Liu, and X. Du, “Analogical reasoning on chinese morphological and semantic relations,” *arXiv preprint arXiv:1805.06504*, 2018.

[38] J. Pennington, R. Socher, and C. Manning, “Glove: Global vectors for word representation,” in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532–1543.