Learn with Noisy Data via Unsupervised Loss Correction for Weakly Supervised Reading Comprehension

Xuemiao Zhang[1], Kun Zhou[3], Sirui Wang[4], Fuzheng Zhang[4], Zhongyuan Wang[4], Junfei Liu[2] *
[1]School of Software and Microelectronics, Peking University, Beijing, China
[2]National Engineering Research Center for Software Engineering, Peking University, Beijing, China
[3]Renmin University of China, Beijing, China
[4]Meituan-Dianping Group
{zhangxuemiao,liujunfei}@pku.edu.cn,francis_kun_zhou@163.com
{wangsirui,zhangfuzheng}@meituan.com,wzhy@outlook.com

Abstract

Weakly supervised machine reading comprehension (MRC) task is practical and promising for its easily available and massive training data, but inevitablely introduces noise. Existing related methods usually incorporate extra submodels to help filter noise before the noisy data is input to main models. However, these multistage methods often make training difficult, and the qualities of submodels are hard to be controlled. In this paper, we first explore and analyze the essential characteristics of noise from the perspective of loss distribution, and find that in the early stage of training, noisy samples usually lead to significantly larger loss values than clean ones. Based on the observation, we propose a hierarchical loss correction strategy to avoid fitting noise and enhance clean supervision signals, including using an unsupervisedly fitted Gaussian mixture model to calculate the weight factors for all losses to correct the loss distribution, and employ a hard bootstrapping loss to modify loss function. Experimental results on different weakly supervised MRC datasets show that the proposed methods can help improve models significantly.

1 Introduction

Machine reading comprehension (MRC) (Rajpurkar et al., 2016) is a well-known NLP task, and has made significant progress in recent years (Yu et al., 2018; Devlin et al., 2019; Gong et al., 2020; Yuan et al., 2020). To learn a well-performed MRC system, large amount of human annotated data is required. However, human annotation is high-cost in real-world application, and it is hard to control the quality for some of hard instances. Recent approach (Joshi et al., 2017) utilized a distantly supervised method to collect the excerpts for answers. It greatly scales up the dataset and reduces the cost, but introduces more harmful noisy samples inevitably. There are many of approaches proposed to filter noise for question answering (QA) recently. Lin et al. (2018) and Lee et al. (2019a) adopted a paragraph selector to calculate confidences of paragraphs to help filter noisy ones before they are input into the main model. Niu et al. (2020) designed a submodel to generate labels to supervise the training of the selector. Back to MRC, Lee et al. (2019b) further proposed to generate labels for unlabeled samples, then train an extra *Refinery* model to refine the overall labels for multilingual MRC task with limited training data.

Admittedly, these multistage methods have achieved certain improvements, but rely heavily on the selector, retriever or refinery. The qualities of these complementary models are hard to be controlled, and make training difficult. In fact, we can explore another novel idea that exploits the essential characteristics of noise itself to help alleviate its effect for MRC task. Inspired by the idea of learning with noisy labels in image classification (Arazo et al., 2019), we explore and find that the loss distribution of weakly supervised MRC training data has inspiring characteristics. As shown in Figure 1 (a), at the beginning of training, losses of noisy samples are generally greater than losses of clean samples significantly. And in Figure 1 (b), during training, the losses of all samples roughly converge into two clusters according to values. In addition, we have noticed that without correction, noise tends to attract more attention due to

*Corresponding author: Junfei Liu (liujunfei@pku.edu.cn)

Proceedings of the 28th International Conference on Computational Linguistics, pages 2624–2634
Barcelona, Spain (Online), December 8-13, 2020

the produced larger loss, which causes the model optimized into wrong direction easily. We argue that it is one of the essential reasons why the performance will be hurt by much noise.

In this paper, our main idea for improving original models is to correct the loss distribution based on the above findings, which reduces losses of noisy samples thus avoiding fitting noise and pushes models to pay more attention to the supervision signals from clean data, as shown in Figure 2. Specifically, instead of modifying the original structures of previous well-performing models, we first choose to fit a 2-component Gaussian mixture model (GMM) to loss distribution unsupervisedly, then infer the probabilities of samples being clean or noisy through the posterior probability provided by GMM. Note that we have verified in Section 5.2 that the noise recognition accuracy can even exceed 90% by GMM. Based on the inferred results of GMM, we then automatically produce weight factors for all losses. Specifically, we assign larger factors to losses that have higher probabilities of being clean samples by GMM, while assign lowers ones to losses that are more likely to be noisy. Note that in the traditional case without correction, the weight factors of all losses can be regarded as an uniform distribution. In addition, we also propose to use the hard bootstrapping loss to replace standard cross-entropy loss to further correct loss values of the individual samples to further avoid fitting noise.

Our contributions are summarized as: (1) We explore the essential characteristics of noise in weakly supervised MRC from the perspective of loss distribution, and offer new ideas for this task and other related NLP tasks in weakly supervised manner; (2) We propose the hierarchical loss correction method to avoid fitting noise and strengthen the supervision from clean samples, which uses unsupervisedly fitted GMM to calculate weight factors for correcting loss distribution, and uses hard bootstrapping loss to modify loss function; (3) We conduct ample experiments on two types of multiple weakly supervised datasets, and experimental results show that the proposed method can improve models significantly.

2 Preliminaries

2.1 Problem Formulation

The typical machine reading comprehension (MRC) task focuses on learning a model $h_\theta(x)$ to answer a question q given the excerpt evidence e derived from excerpt set $\mathcal{E}$. The training set can be formalized into a set of triple examples $\mathcal{D} = \{(q_i, e_i, a_i)|i = 1, ..., N\}$, where N is the number of examples in $\mathcal{D}$, $q_i = \{w_1^{q_i}, w_2^{q_i}, \ldots, w_n^{q_i}\}$ is the question with n tokens, $e_i = \{w_1^{e_i}, w_2^{e_i}, \ldots, w_m^{e_i}\}$ is the excerpt evidence with m tokens, $a_i = \{w_i^{e_i}, w_{i+1}^{e_i}, \ldots, w_{i+s-1}^{e_i}\}$ is a substring from e_i, and defines the golden answer to q_i. Following Devlin et al. (2018) and Joshi et al. (2017), this task can be formulated as to predict an answer span, i.e., the start and end indices of answer a_i in excerpt e_i.

TriviaQA (Joshi et al., 2017) contains a distantly supervised MRC dataset, whose evidences are gathered automatically, with the assumption of distant supervision that the presence of the answer string in an evidence document implies that the document does answer the question. Formally, in the distantly supervised MRC task, e_i is set to a set of excerpts, and training data is formalized as $\mathcal{D}_{ds} = \{(q_i, \{e_{ij}\}_{j=1}^{M}, a_i)|i = 1, ..., N\}$, where M is the number of excerpts. Although all excerpts in the set contain answer strings, there is no guarantee that answers to questions will be derived from the excerpts. When aligned to standard MRC data, a sample of distant supervised data $(q_i, \{D_i^1, D_i^2, \ldots, D_i^M\}, a_i)$ can be expanded into M samples in standard format $\{(q_i, D_i^1, a_i), (q_i, D_i^2, a_i), \ldots, (q_i, D_i^M, a_i)\}$. Obviously this automated operation can easily obtain a large number of training data, but inevitably introduces a lot of noise, which will hurt the model's performance.

In this paper, we consider such a more common and general weakly supervised MRC scenario, which extends from the distantly supervised MRC task (Joshi et al., 2017): in the training set $\mathcal{D}$, both the excerpts and the answer spans may be noisy. That is, not only do the excerpt e_i not guaranteed to provide the evidence to answer the question q_i, but the answer span a_i itself is likely to be noise. Anyway, $x_i \in \mathcal{D}$ is a noisy sample when excerpt evidence e_i or answer span a_i is noisy. We focus on improving the models on weakly supervised MRC training data.

2.2 Empirical Explorations

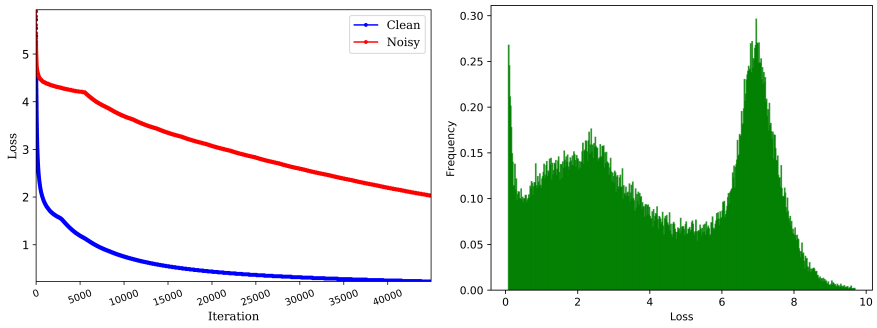
Typical MRC models usually learn the model parameters θ by minimizing the following loss function:

$$\mathcal{L} = -\sum_{i=1}^{N} log(P_{s_i}^1) + log(P_{e_i}^2) = -\sum_{i=1}^{N} y_i^T \log(P(a_i|e_i, q_i)) = -\sum_{i=1}^{N} y_i^T \log(h_\theta(x_i)) \quad (1)$$

where s_i and e_i of answer a_i are the start and end positions in excerpt e_i for sample x_i. $P_{s_i}^1$ and $P_{e_i}^2$ are the probabilities of the starting and ending position, respectively. y_i defines the label of the start and end indices. $h_\theta(x)$ defines the softmax probability produced by the model.

Taking Eq. (1) as the loss function, we train MRC models on weakly supervised datasets and record the entire loss convergence process, and collect all samples' losses computed by a trained model instance, as shown in Figure 1. From Figure 1(a), we can find that in the early stages of training, noise samples usually lead to significantly larger losses than clean samples. And from Figure 1(b), the losses of the entire dataset can be roughly divided into two clusters, we argue that the cluster with larger mean loss value corresponds to noisy samples, and conversely the other corresponds to clean samples.

These observations intuitively suggest that we can use a 2-component mixture model to unsupervisedly fit the overall loss distribution, where two independent components correspond to the loss distributions caused by noise and clean data, respectively. During training, we can reasonably correct the loss distribution before the loss back propagation by using the mixture model to infer whether the losses come from noise or clean data, thereby reducing disturbance from noise and pushing the model to pay more attention to the supervision signals from clean data. It is worth noting that the entire process does not use any additional supervision signals, but it gives the model much additional important information.



(a) (b)

Figure 1: Analysis of loss characteristics. (a): Comparison of loss convergence processes when training on original SQuAD data and noisy SQuAD data with 80% noise; (b): Frequency distribution histogram of losses obtained by infering all samples of distantly supervised TriviaQA data using a model instance.

3 Methodology

We propose hierarchical loss correction strategy to avoid fitting noise and enhance supervision signals from clean samples. The overall framework of the proposed methods is shown in Figure 2. We first *model loss* by fitting a GMM, then perform *loss correction* operation before back propagation.

3.1 Modeling Loss

Based on observations in Section 2.2, we can effectively infer whether a sample is more likely to be clean or noisy by fitting a probability distribution model to the losses of all training data. Intuitively, we argue that losses corresponding to clean and noisy samples obey two independent probability distributions, respectively. Therefore, losses of all training samples obey a mixture probability distribution composed of the above two distributions. We employ the widely used unsupervised GMM to fit the losses, since

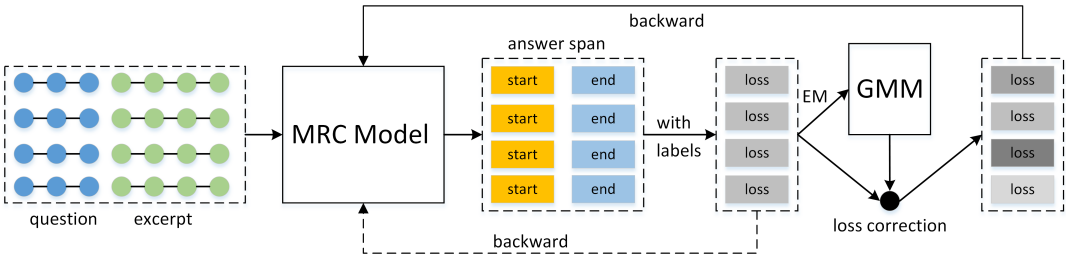


Figure 2: The framework of the proposed methods. Original losses are computed by aligning the predictions with the ground truth. A GMM is fitted to them to give the posterior probabilities to compute the corrected loss distribution for actual back propagation. original losses are used during pretraining.

loss histogram in Figure 3 shows Gaussian distribution is suitable, which has good mathematical properties. Specifically, we use 2-component GMM to fit the loss distributions of clean and noisy samples, respectively. Next, we introduce how to fit GMM to losses unsupervisedly and use it to model noise.

We assume that the observed losses $\boldsymbol{l} = \{l_i\}_{i=1}^{N}$ can be generated by a GMM θ_G:

$$P(\boldsymbol{l}|\theta_G) = \sum_{i=1}^{K} \alpha_k \phi(\boldsymbol{l}|\theta_k) \tag{2}$$

where $\theta_G = (\alpha_1, \alpha_2, \ldots, \alpha_K; \theta_1, \theta_2, \ldots, \theta_K)$, θ_k are parameters of the k-th Gaussian component and α_k are mixing coefficients for the convex combination of each individual probability density function (PDF) $p(\boldsymbol{l}|\theta_k)$. We employ the Expectation Maximization (EM) algorithm to fit GMM to the observed losses.

Specifically, we define the latent variables $\hat{\gamma}_{jk}$ to be the posterior probability of the point l_j having been generated by mixture component θ_k, where $j = 1, 2, \ldots, N, k = 1, 2, \ldots, K$. In the E-step we fix the parameters α_k, θ_k and update the latent variables using Bayes rule:

$$\hat{\gamma}_{jk} = E(\gamma_{jk}|\boldsymbol{l}, \theta_G) = P(\gamma_{jk} = 1|\boldsymbol{l}, \theta_G) = \frac{\alpha_k \phi(l_j|\theta_k)}{\sum_{k=1}^{K} \alpha_k \phi(l_j|\theta_k)} \tag{3}$$

And given fixed $\hat{\gamma}_{jk}$, the M-step estimates parameters $\hat{\mu}_k$, $\hat{\delta}_k$ of the Gaussian distribution, and $\hat{\alpha}_k$ as:

$$\hat{\mu}_k = \frac{\sum_{j=1}^{N} \hat{\gamma}_{jk} l_j}{\sum_{j=1}^{N} \hat{\gamma}_{jk}}; \hat{\sigma}_k = \frac{\sum_{j=1}^{N} \hat{\gamma}_{jk}(l_j - \mu_k)^2}{\sum_{j=1}^{N} \hat{\gamma}_{jk}}; \hat{\alpha}_k = \frac{\sum_{j=1}^{N} \hat{\gamma}_{jk}}{N} \tag{4}$$

Repeat the above calculation until convergence or the iterations exceeds the maximum limitation.

Given a fitted GMM, we can effectively model the losses. Specifically, we calculate the probability of a sample being clean or noisy through the posterior probability as follows:

$$p(\theta_k|l_i) = \frac{p(\theta_k)p(l_i|\theta_k)}{p(l_i)} \tag{5}$$

We use the component θ_k with the smallest mean μ_k to represent the loss distribution of clean samples.

3.2 Hierarchical Loss Correction

We further consider correcting losses to avoid fitting noise. The correction process includes hierarchical operations, fine-grained loss function correction and high-level loss distribution correction.

Since standard cross-entropy (CE) in loss Eq. (1) is ill-suited to deal with noisy samples because the model will exploit wrong knowledge from noisy samples (Zhang et al., 2017), it is replaced with the hard bootstrapping loss (Reed et al., 2015) to correct the training objective and alleviate the disturbance of noise, which deals with noisy samples by adding a perception term to CE loss:

$$\mathcal{L}_{hard} = -\sum_{i=1}^{N} (\beta y_i + (1-\beta) z_i)^T) log(h_i) \tag{6}$$

where $z_i := \mathbb{1}[k = \arg\max h_j, j = 1, \ldots, N]$, β weights the model prediction z_i in the loss function. Following Reed et al. (2015), we set $\beta = 0.8, \forall i$.

We further propose to correct the loss distribution based on the posterior probability by GMM. Generally, neural MRC models are trained by stochastic gradient descend (SGD) approach, in which losses directly affect the calculation of gradients, which in turn affect the optimization process, so that samples with larger losses have more influence. Traditional models trained on clean data try to fit all losses with the intuition that the under-fitting leads to large losses. But when training on noisy data, we argue large losses are more likely to be caused by noise and need to be corrected. We correct entire loss distribution by using GMM to infer the possibilities that samples are clean, and adopting a softmax operation to assign larger weight factors to the samples with higher probabilities and lower ones to others. The loss distribution correction operation with weight factors is given as:

$$\mathcal{L}_{correct} = \sum\nolimits_{i=1}^{N} \frac{1}{Z} e^{\frac{p(k=k_c|l_i^{hard})}{T}} l_i^{hard} \tag{7}$$

where $Z = \sum_{j=1}^{N} e^{\frac{p(k=k_c|l_j^{hard})}{T}}$ is the normalization factor, $k_c = \arg\min(\theta_G.means)$ is the Gaussian component with the smallest value of mean parameter in GMM model θ_G, indicating that it is clean component fitted to the clean data, and T is the temperature parameter.

Algorithm 1: Loss correction process for reading comprehension question answering.

Input: Training epoch number K; training data size N; train triple samples $\{x_i\}_{i=1}^{N}$; GMM refitting frequency f; the size of mini-batch b.

initialize MRC model θ;
Pretrain θ with original losses by standard cross entropy;
for $k \leftarrow 1$ *to* K **do**
  if $k \ \% \ f == 0$ **then**
    Compute all losses $\boldsymbol{l}$ of all samples $\{x_i\}_{i=1}^{N}$ by Eq. (6);
    Fit GMM θ_G to all losses $\boldsymbol{l}$ using EM algorithm as Eq. (3) and Eq. (4);
    $k_c \leftarrow \arg\min(\theta_G.means)$ // choosing the Gaussian component with the smallest mean value to represent the distribution of clean data;
  for *mini-batch in batches of epoch* **do**
    Compute batch losses $\boldsymbol{l}_{hard}$ of the mini-batch samples $\{x_i\}_i^b$ by Eq. (6);
    Compute posterior probabilities $\{p(k = k_c|l_i)\}_{i=1}^{b}$ for $\boldsymbol{l}_{hard}$;
    Compute corrected batch losses $\boldsymbol{l}_{hard}^{c}$ by Eq. (7);
    Loss back propagation from $\boldsymbol{l}_{hard}^{c}$ and updata θ;

3.3 Overviews

In summary, the framework of the proposed methods is shown in Figure 2, and we train the improved moels according to Algorithm 1. In practice, we first pretrain the original model using standard CE. Then, we compute the bootstrapping losses, and fit a 2-component GMM to these losses using EM algorithm and record the clean Gaussian component with minimum mean value. In each training step, we compute batch losses of the batch samples and the probabilities of these samples being clean, then employ a softmax operation to compute the weight factors to further calculate the corrected losses. At the end of the step, we do back propagation based on the corrected losses.

4 Experimental Setup

4.1 Datasets

SQuAD. SQuAD (Rajpurkar et al., 2016) is a standard and high-quality MRC dataset. The annotators were asked to write more than 100,000 questions and select a span of arbitrary length from the given Wikipedia paragraph to answer the question. In practice, we use the SQuAD v1.1, and randomly select a certain percentage of samples to add noise to them. For each noisy sample, we randomly select a

continuous sequence of tokens from the evidence paragraph to replace the original label. Note that in this scenario, the answer is noisy. In order to fully explore the influence of noise, we generate 4 noisy training data, and their noise ratios are 0.2, 0.4, 0.6 and 0.8, respectively.

TriviaQA. TriviaQA (Joshi et al., 2017) is a collection of trivia question-answer pairs that were scraped from the web. We use their distantly supervised MRC dataset whose excerpt evidences are scraped from Wikipedia. We convert TriviaQA into a weakly supervised data format that conforms to the definition in the section 2.1. Note that, in this scenario, the evidence file is noisy. However, unlike the randomly created noise in squad, noise in TriviaQA is real in natural scenes.

4.2 Setup

Baselines. We use two widely used models (Cui et al., 2019; Lee et al., 2019b), and a shrunken model as the baselines. **BERT**: We modify a pre-trained uncased BERT (Devlin et al., 2018) model on a masked language task to MRC task by mapping the features extracted by BERT into the inferencing position logits to predict answer spans through a dense layer. **BiDAF**: Seo et al. (2016) proposed a multistage hierarchical process, which represents context at different levels of granularity, and uses a two-way attention flow mechanism to obtain query-aware context representation, we follow the implementation setting of original BiDAF. **BiDAF**$_m$: To explore the impact of model capacity on the proposed methods, we build a mini version of BiDAF, denoted BiDAFm, by reducing the amount of parameters; specifically, we set word dimension to 50 (original 100), char channel size to 20 (original 100), hidden size of LSTM to 35 (original 100), char channel width to 2 (original 5) and char dimension to 3 (original 8).

Evaluation Metrics. Following Chen et al. (2017) and Lee et al. (2019b), we use these two official evaluation metrics to evaluate our models, namely ExactMatch (EM) and F1 score. Among them, EM evaluates the percentage of prediction answers that exactly match one of the ground truth ones and F1 score can measure the average overlap between the prediction and ground truth answer. And we directly use the official evaluation script provided by SQuAD v1.1 for evaluation.

Settings. We implement the proposed methods by employing the loss correction strategies based on the above three baselines, including using a mixture probability distribution model to fit to losses of models, which in turn helps correct the loss distribution, and replacing the cross-entropy loss in Eq. (6) to the hard bootstrap loss which is more suitable for processing noisy data. Based on these settings, we retrain these new models in the same experimental environment. In practice, for mixture models, we use 2-component GMM, and its max iteration number is set to 100. We use Glove pretrained embeddings to initialize word embedding in BiDAF. We set β in hard bootstrapping loss to 0.8, set learning rate in BERT and BiDAF to 0.0005 and 0.001, respectively, and set temperature T to 1.0. We bounding the loss observations in $[\epsilon, 1-\epsilon]$ instead of $[0, 1]$ ($\epsilon = e - 4$ in practice) to sidesteps this issue that EM algorithm will become numerically unstable when the observations are very near 0 and 1.

5 Results and Analysis

5.1 Experimental Results

Table 1 shows the evaluation results of the baselines and the improved models using the proposed methods on EM and F1 metrics. We can find that our methods make the original well-performed models achieve a further significant performance improvement on the real distantly supervised TriviaQA dataset. Among them, the improved model based on BERT improves by 13.9% and 10.0% on the EM and F1 respectively, and the improved model based on BiDAF$_m$ improves by 17.4% and 13.2%, respectively. It shows that the proposed methods can effectively improve the models training on noisy data. On noisy SQuADs with different ratios of noise, our methods can still significantly improve models. Taking SQuAD with 60% noise as an example, the improved model based on BiDAF has improved 10.42 percentage points (29.4%) and 9.50 points (21.1%) on EM and F1, respectively. The improved model based on BERT has improved 8.07 percentage points (20.6%) and 8.20 points (16.6%), respectively. It shows that the proposed methods can indeed help reduce the disturbance of noise on the model, and this ability can be clearly reflected on different data sets.

Model		SQuAD										TriviaQA	
		clean		noise-0.2		noise-0.4		noise-0.6		noise-0.8			
		EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1
BiDAF_m	OR	60.19	71.87	58.13	69.53	54.08	65.99	38.92	50.79	5.07	7.38	17.41	22.24
	HB	-	-	58.50	70.23	54.47	66.21	42.81	52.05	6.40	8.92	19.22	23.32
	DCE	-	-	**59.09**	70.28	55.93	66.94	**47.23**	**57.89**	8.25	10.84	19.86	24.45
	DHB	-	-	58.61	**70.47**	**56.25**	**67.54**	43.71	52.95	**8.52**	**11.09**	**20.44**	**25.18**
BiDAF	OR	64.18	74.70	60.02	70.62	53.42	63.85	35.36	44.95	10.35	15.35	22.92	27.23
	HB	-	-	61.94	72.01	57.35	67.58	34.91	44.89	10.63	15.98	23.00	27.17
	DCE	-	-	63.25	73.15	57.75	68.46	**45.78**	**54.45**	11.14	15.97	**23.17**	**27.41**
	DHB	-	-	**63.36**	**73.89**	**59.13**	**70.38**	43.91	53.01	**12.16**	**17.12**	23.14	27.28
BERT	OR	69.56	79.08	61.36	72.48	53.13	64.10	39.08	49.44	15.49	24.60	25.65	30.95
	HB	-	-	62.37	73.16	54.22	64.82	43.74	54.03	17.75	25.84	26.24	31.70
	DCE	-	-	63.06	73.73	54.99	66.09	43.73	53.48	17.36	**26.62**	28.28	33.34
	DHB	-	-	**64.12**	**74.09**	**56.94**	**67.34**	**47.15**	**57.64**	**18.43**	26.09	**29.21**	**34.02**

Table 1: Evaluation results of different models under different loss correction strategies on two category of weakly supervised training sets. Among them, *OR* represents the original methods with cross entropy, *HB* represents methods using hard bootstrapping loss only, and *DCE* and *DHB* represent strategies of using loss distribution correction based on cross entropy and hard bootstrap loss, respectively.

Ablation Study. For each group of experiments, we report the experimental results of different models using original cross entropy loss and hard bootstrapping loss, and using high-level loss distribution correction with the two loss functions, respectively. From Table 1, we can find that: (1) Compared with using original cross entropy, the strategy of only correcting the loss function with hard bootstrapping loss can also improve models to a certain extent. (2) Both loss correction combination strategies have significant impacts on models' promotions. (3) The models using the loss distribution correction based on standard cross-entropy strategy has been effectively improved compared to the baselines, and some models using this strategy perform best in some scenarios, such as the BiDAF-based improved model trained on SQuAD with 60% noise. (4) But overall, the improved models using loss distribution correction based on hard bootstrap loss strategy will perform better, because the strategy attempts to provide cleaner loss signals by correcting both the loss values of the samples themselves and the loss distribution. Since there is no guarantee that adopting the combination strategy based on hard bootstrap loss will be better, we recommend to try both combination strategies if conditions permit, and choose the one that performs better, in the practice of applying the proposed methods.

5.2 How does GMM work?

We further analyze GMM's ability to distinguish between noisy and clean samples based on loss distribution unsupervisedly. First, independent of the noisy SQuAD sets for training, we randomly regenerate a series of test sets from original training set to evaluate GMM, which contain corresponding proportions noise and labels used to mark whether the samples are noise. Specifically, we regularly use the model in normal training process to output the loss corresponding to each sample in the corresponding test set, and use a new GMM instance to fit this loss distribution. Then use the fitted GMM to infer whether the sample is clean or noise. Along with training process, we record the best evaluation results of GMM.

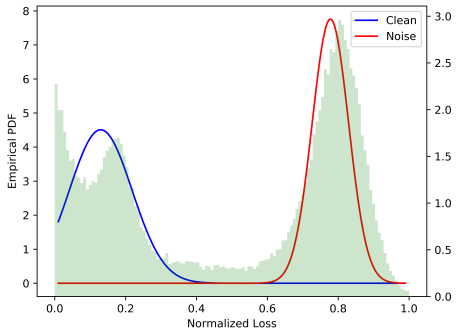
From Table 2, we can find that GMM can very effectively identify noise. On data sets with a noise ratio of 60% or less, BERT-based and BiDAF-based improved models can correctly identify more than 97% and 80% of noisy samples, respectively. And on the noisy data a noise ratio of 80%, the noise recognition rate still reaches 74%. This means that based on the observations in Section 2.2, GMM can provide so much extra useful information out of nothing to help improve the models. Specifically, the posterior probability given by GMM help to correct the loss distribution, thereby reducing the disturbance of noise, and push the model pay more attention to the supervision signals from clean data. We also note that the recognition rates of noise and clean data is a trade-off. Noise recognition and clean recognition are difficult to perform both well at the same time. However, in general, the recognition results of GMM are very effective in correcting the loss distribution, because as long as the attentions to clean samples are increased or the to noisy samples are reduced, the model can be optimized in a more correct direction.

Model		noise-0.2			noise-0.4			noise-0.6			noise-0.8		
		all	noise	clean	all	noise	clean	all	noise	clean	all	noise	clean
BiDAF$_m$	OR	74.30	99.62	67.95	87.24	79.59	92.30	56.57	81.96	18.21	32.26	17.29	92.11
	DHB	54.54	99.77	43.21	87.50	79.23	92.96	72.52	80.20	60.91	33.62	17.20	99.31
BiDAF	OR	78.64	99.61	73.38	81.27	98.61	69.81	77.94	81.93	71.91	72.33	81.81	34.45
	DHB	60.47	99.82	50.60	87.76	80.12	92.81	78.17	81.21	73.58	30.31	17.91	79.90
BERT	OR	72.00	99.35	65.14	67.20	98.97	46.24	68.74	97.02	26.08	71.22	74.16	59.45
	DHB	76.71	99.62	70.96	72.78	98.86	55.58	76.23	98.27	42.95	68.07	74.74	41.36

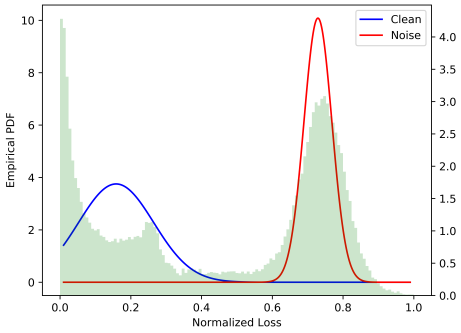
Table 2: Accuracy of unsupervisedly identifying the noise in the training data of different noisy SQuAD with different noise rates by GMM obtained by fitting to the loss observations. Among them, *all* represents the overall accuracy, *noise*, and *clean* respectively are the proportion of noise samples and clean samples that are correctly identified.

5.3 Fit to Loss Distribution

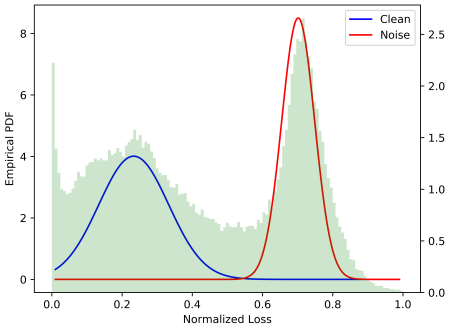
In addition, we intuitively show how GMM fits the loss distribution, as shown in Figure 3. From Figure 3, we can find that the loss distribution of different models trained on different noisy data sets can be indeed roughly divided into two clusters, indicates that it makes sense to use a two-component mixture probability model to fit the loss distribution. Moreover, the Gaussian distribution is very universal, because it can basically fit loss clusters in various situations. Of course, the operators can explore or design a special distribution to replace the Gaussian distribution for specific scenarios in practice. Note that we focus more on the generalization ability of the Gaussian distribution in this paper.



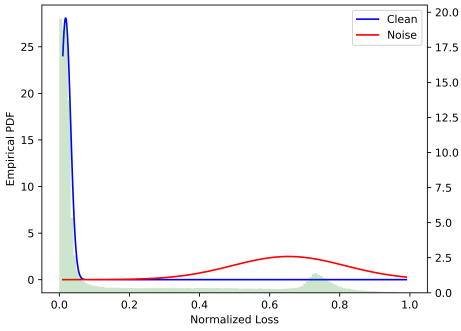
(a) BiDAF on SQuAD with 60% noise



(b) BERT on SQuAD with 60% noise



(c) BiDAF on TriviaQA



(d) BERT on TriviaQA

Figure 3: Analysis of fitting a 2-component Gaussian mixture model to the losses computed by models (BiDAF and BERT) on different noisy data sets, where two clusters in the histogram correspond to two Gaussian components depicted by the red and blue curves, respectively.

5.4 Explore Other Mixture Model

From observations in Section 2.2, we can know that as long as a probability model can well fit the loss distribution, it can be used to participate in the construction of the mixture model. In addition to GMM, we also explore the Beta Mixture Model (BMM), which performs well in noisy image classification

Model	PDM	SQuAD								TriviaQA	
		noise-0.2		noise-0.4		noise-0.6		noise-0.8			
		EM	F1	EM	F1	EM	F1	EM	F1	EM	F1
BiDAF$_m$	BMM	58.18	69.59	53.93	66.63	**47.39**	57.31	6.42	9.52	19.20	24.77
	GMM	**58.61**	**70.47**	**56.25**	**67.54**	47.23	**57.89**	**8.52**	**11.09**	**20.44**	**25.18**
BERT	BMM	62.89	73.75	55.19	65.85	43.27	51.84	**18.97**	**28.93**	27.38	32.41
	GMM	**64.12**	**74.09**	**56.94**	**67.34**	**47.15**	**57.64**	18.43	26.09	**29.21**	**34.02**

Table 3: Comparison results of employing different mixture models to improve BiDAF$_m$ and BERT on different noisy data sets.

tasks (Arazo et al., 2019). The beta distribution over a normalized loss $l \in [0, 1]$ is defined to have PDF: $p(l|\alpha, \beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} l^{\alpha-1}(1-l)^{\beta-1}$, where $\Gamma(\cdot)$ is the Gamma function, and $\alpha, \beta > 0$ are parameters. Similarly, the mixture PDF is given by substituting the above into Eq. (5). Based on BiDAF$_m$ and BERT, we conduct comparison experiments on all noisy datasets. The experimental results are shown in Table 3. From Table 3, we can find that: (1) loss correction based on BMM can also bring a significant performance improvement, compared with the results in Table 1; (2) in most scenarios, GMM can help to achieve more significant improvements than BMM, indicating that GMM has obvious advantages in MRC task, and is very suitable for this task. It enlightens us that when there is no better choice, the Gaussian mixture model is a good solution, or serves it as a baseline to explore better models.

6 Related Work

Machine Reading Comprehension. Machine reading comprehension (MRC) (Rajpurkar et al., 2016) has received increasing attention recently, which requires a model to extract an answer span to a question from reference documents (Yu et al., 2018; Devlin et al., 2019; Liu et al., 2020; Zheng et al., 2020; Yuan et al., 2020). Owing to the rise of pre-training models (Devlin et al., 2018), a machine is able to achieve highly competitive results on classic datasets (*e.g.* SQuad (Rajpurkar et al., 2016)), even close to human performance. However, there is still a huge gap between high performance on the leaderboard and poor practical user experience, due to the noisy dataset, high-cost annotation and low resource languages. Recently, the more challenging distantly supervised MRC task, TriviaQA (Joshi et al., 2017) was proposed, in which the provided evidences are noisy and collected based on the distant supervision. (Yuan et al., 2020) proposed a multilingual MRC task to facilitate the study on low resource languages. (Lee et al., 2019b) focused on annotating the unlabeled data with heuristic method and refine the labels by an extra *Refinery* model for multilingual MRC task.

Learning with Noisy Labels. Recently, the great progress has been made on learning with noisy labels in image classification and question answering (QA) domains. Reed et al. (2015) and Ma et al. (2018) proposed a bootstrapping method to reconstruct loss function for noisy data combined with model predictions. Jiang et al. (2018) and Arazo et al. (2019) put forward an empirical assumption that samples with lower losses are clean, then separate the clean and noisy samples based on the loss distribution. For QA task, Lin et al. (2018) and Lee et al. (2019a) utilized an extra paragraph selector to filter noise by calculating confidences of paragraphs. Niu et al. (2020) further proposed a complementary model to generate labels to the paragraphs for training selectors supervisedly.

7 Conclusion

In this paper, we explore natural characteristics of noise from perspective of loss, and find in early stages of training, noisy samples usually result in significantly larger losses than clean samples. Based on the observation, we propose a hierarchical loss correction strategy to avoid fitting noise and strengthen supervision signals from clean samples by incorporating an unsupervisedly fitted GMM and modifying original loss funciton to hard bootstrapping loss. We conducted ample experiments on multiple weakly supervised MRC datasets. Experimental results show that the proposed methods can effectively help models to achieve significant improvements.

References

Eric Arazo, Diego Ortego, Paul Albert, Noel E O'Connor, and Kevin McGuinness. 2019. Unsupervised label noise modeling and loss correction. In *International Conference on Machine Learning (ICML)*, pages 312–321, June.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879, Vancouver, Canada, July. Association for Computational Linguistics.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Shijin Wang, and Guoping Hu. 2019. Cross-lingual machine reading comprehension. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1586–1595, Hong Kong, China, November. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

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, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Hongyu Gong, Yelong Shen, Dian Yu, Jianshu Chen, and Dong Yu. 2020. Recurrent chunking mechanisms for long-text machine reading comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6751–6761, Online, July. Association for Computational Linguistics.

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 Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2304–2313, Stockholmsmässan, Stockholm Sweden, 10–15 Jul. PMLR.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019a. Latent retrieval for weakly supervised open domain question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6086–6096, Florence, Italy, July. Association for Computational Linguistics.

Kyungjae Lee, Sunghyun Park, Hojae Han, Jinyoung Yeo, Seung-won Hwang, and Juho Lee. 2019b. Learning with limited data for multilingual reading comprehension. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2840–2850, Hong Kong, China, November. Association for Computational Linguistics.

Yankai Lin, Haozhe Ji, Zhiyuan Liu, and Maosong Sun. 2018. Denoising distantly supervised open-domain question answering. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1736–1745, Melbourne, Australia, July. Association for Computational Linguistics.

Dayiheng Liu, Yeyun Gong, Jie Fu, Yu Yan, Jiusheng Chen, Daxin Jiang, Jiancheng Lv, and Nan Duan. 2020. RikiNet: Reading Wikipedia pages for natural question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6762–6771, Online, July. Association for Computational Linguistics.

Xingjun Ma, Yisen Wang, Michael E. Houle, Shuo Zhou, Sarah M. Erfani, Shu-Tao Xia, Sudanthi N. R. Wijewickrema, and James Bailey. 2018. Dimensionality-driven learning with noisy labels. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 3361–3370. PMLR.

Yilin Niu, Fangkai Jiao, Mantong Zhou, Ting Yao, jingfang xu, and Minlie Huang. 2020. A self-training method for machine reading comprehension with soft evidence extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3916–3927, Online, July. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas, November. Association for Computational Linguistics.

Scott Reed, Honglak Lee, Dragomir Anguelov, Christian Szegedy, Dumitru Erhan, and Andrew Rabinovich. 2015. Training deep neural networks on noisy labels with bootstrapping. In *International Conference on Learning Representations (ICLR)*.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2016. Bidirectional attention flow for machine comprehension. *arXiv preprint arXiv:1611.01603*.

Adams Wei Yu, David Dohan, Quoc Le, Thang Luong, Rui Zhao, and Kai Chen. 2018. Fast and accurate reading comprehension by combining self-attention and convolution. In *International Conference on Learning Representations*.

Fei Yuan, Linjun Shou, Xuanyu Bai, Ming Gong, Yaobo Liang, Nan Duan, Yan Fu, and Daxin Jiang. 2020. Enhancing answer boundary detection for multilingual machine reading comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 925–934, Online, July. Association for Computational Linguistics.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. 2017. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations (ICLR)*.

Bo Zheng, Haoyang Wen, Yaobo Liang, Nan Duan, Wanxiang Che, Daxin Jiang, Ming Zhou, and Ting Liu. 2020. Document modeling with graph attention networks for multi-grained machine reading comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6708–6718, Online, July. Association for Computational Linguistics.