Instance-adaptive training with noise-robust losses against noisy labels

Lifeng Jin Linfeng Song Kun Xu Dong Yu
Tencent AI Lab
Bellevue, WA, USA
(lifengjin,lfsong,kxkunxu,dyu)@tencent.com

Abstract

In order to alleviate the huge demand for annotated datasets for different tasks, many recent natural language processing datasets have adopted automated pipelines for fast-tracking usable data. However, model training with such datasets poses a challenge because popular optimization objectives are not robust to label noise induced in the annotation generation process. Several noise-robust losses have been proposed and evaluated on tasks in computer vision, but they generally use a single dataset-wise hyperparamter to control the strength of noise resistance. This work proposes novel instance-adaptive training frameworks to change dataset-wise hyperparameters of noise resistance in such losses to be instance-specific. Such instance-specific noise resistance hyperparameters are predicted by special instance-level label quality predictors, which are trained along with the main models. Experiments on noisy and corrupted NLP datasets show that proposed instance-adaptive training frameworks help increase the noise-robustness provided by such losses, promoting the use of the frameworks and associated losses in training NLP models with noisy data.

1 Introduction

The wide availability of neural network models has allowed development of novel and complex natural language processing tasks, many of which are in low-resource settings. With new definitions of tasks comes challenges of constructing new datasets, which is still an expensive and time-intensive endeavor. Many researchers have resorted to constructing datasets by using completely automated pipelines (e.g. Lan et al., 2017; Joshi et al., 2017; Paul et al., 2019; Lange et al., 2019; Sousa et al., 2019; Wu et al., 2020). However, silver labels collected this way are still quite noisy compared to expert annotation. Because such methods have been gaining popularity and practicality, it is important to explore ways to ensure good performance in spite of noisy labels in training data.

The widely-used cross entropy (CE) loss as the optimization objective in classification tasks has been shown to overfit to label noise (Ghosh et al., 2017). Several noise-robust losses have been designed for training models with noisy labels (Reed et al., 2015; Zhang and Sabuncu, 2018; Wang et al., 2019c), which were a convenient way to address the noisy label issue and shown to be more robust than CE. Experiments are usually conducted on computer vision datasets such as CIFAR (Krizhevsky, 2009) and MNIST (LeCun et al., 1998).

These noise-robust losses usually have hyperparameters for determining the strength of the noise-robustness at the dataset level. However, individual training instances may have different amounts of noise, derived from biases within models used in the automated pipeline. Moreover, noisy labels in natural language datasets potentially pose a greater challenge because instances of the same true label may not share similar surface features. Therefore, this work focuses on the improvement of training with noisy labels using noise-robust losses in NLP. We propose two robust training frameworks where the noise-robustness hyperparameters are instance-specific. They are predicted by label quality predictors, which are trained either jointly or iteratively with main models in order to take advantage of any correlation between label quality and input features. Such frameworks are tested with many noise-robust losses on several noisy and corrupted NLP datasets. Results from experiments show that:

1. Instance-adaptive noise-robust training proposed in this work enhances the noise-robustness of the losses on noisy and corrupted datasets, which results in large performance gains when instance-specific noise-resistance hyperparameters are used.

2. Noise-robust losses are an effective way to

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 5647–5663
November 7–11, 2021.

combat noise in silver-standard NLP datasets, especially when the noise rate is high. ER-GCE loss proposed in this work achieves the best performance on all datasets compared the noise-robust losses from previous work.

2 Noise-robust losses

We first define a dataset $\mathcal{D}$ for single-label classification as a tuple of input features and corresponding labels $\{\mathbf{x}_i, y_i\}_{i=1}^N$, where $y_i \in \{1, ..., K\}$ is the annotated label and $\mathbf{y}_i \in \{0, 1\}^K$ is a one-hot representation of the annotated label with K total possible classes for training instance i. Given a classification model f with trainable parameters $\boldsymbol{\theta}$, the predicted conditional distribution of the classes from the model is $\mathbf{d}_i = f(\mathbf{x}_i)$. Training the model f is then trying to find the set of parameters $\boldsymbol{\theta}^*$ which minimizes the empirical risk $\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \sum_{i=1}^N L(f(\mathbf{x}_i), \mathbf{y}_i)$, with L being a loss function which takes the model output and the annotated label, and returns a non-negative value. CE is the commonly used loss for classification, which is defined as the negative log-likelihood of the annotated class $\text{CE}(\mathbf{y}, \mathbf{d}) = -\mathbf{y}^\top \log f(\mathbf{x})$.

Theoretical results (Du Plessis et al., 2014; Ghosh et al., 2017) have shown that losses which satisfy

$$\sum_{k=1}^{K} L(f(\mathbf{x}), \bar{\mathbf{y}}_k) = C, \forall \mathbf{x} \in \mathcal{D}, \forall f, \quad (1)$$

with C being some constant and $\bar{\mathbf{y}}_k$ being a one-hot representation of a label at k, are robust against symmetric and label-dependent noise with noise rate $\eta < \frac{K-1}{K}$, which is the probability that the annotated label y is not the true label $\hat{y}$. However, for losses which cannot satisfy this condition where the sum of loss values with respect to all classes is constant, they are more noise-robust if the above term is bounded instead of unbounded. Examples of each condition include CE being unbounded, mean squared error being bounded and mean absolute error (MAE) being constant.

2.1 Overview of noise-robust losses

Many noise-robust losses have been proposed and evaluated, mostly on vision datasets. The noise-robust losses that are examined in this work include the soft and hard variants of the bootstrapping loss (BSL, Reed et al., 2015), generalized cross entropy (GCE, Zhang and Sabuncu, 2018), symmetric cross entropy (SCE, Wang et al., 2019c), a new loss – entropy-regularized general cross entropy (ER-GCE), and two baselines based on simple modifications of CE – weighted cross entropy (WCE) and label smoothing (LS, Szegedy et al., 2016). These noise-robust losses are formulated below with a hyperparameter β which is negatively correlated with the noise-robustness. When β approaches 1, they become the least noise-robust but have fast convergence. When β approaches 0, they become the most noise-robust, but may underfit the training data (Wang et al., 2019c).

Weighted corss entropy (WCE): One simple way to use CE to combat noise is to apply weights to different training instances according to their quality:

$$\text{WCE}(\mathbf{y}, \mathbf{d}) = -\beta \mathbf{y}^\top \log f(\mathbf{x}), \quad (2)$$

where β is a noise-robustness hyperparameter. With a dataset-specific β, WCE is equivalent to CE with no noise-robustness. Noise-robustness may be achieved when each training instance $\mathbf{x}_i$ gets a β_i, as described in Section 3.

Label smoothing (LS): Another simple way to use CE to combat noise is to convert the one-hot targets into soft targets:

$$\text{LS}(\mathbf{y}, \mathbf{d}) = -\mathbf{y}_{\text{LS}}^\top \log f(\mathbf{x}), \quad (3)$$

$$\mathbf{y}_{\text{LS}} = \beta \mathbf{y} + (1 - \beta)/K, \quad (4)$$

where β controls how smooth a target is.

Bootstrapping loss (BSL): BSL combines two components in the loss: the distance to the noisy training target, which is measured by CE, and model confidence of its predictions, which is measured by the entropy of model prediction $H(\mathbf{d})$. The soft BSL is the sum of both terms:

$$\text{BSL}_s(\mathbf{y}, \mathbf{d}) = -\beta \mathbf{y}^\top \log \mathbf{d} + (1 - \beta) H(\mathbf{d}). \quad (5)$$

For the hard BSL, the entropy function is replaced by max:

$$\text{BSL}_h(\mathbf{y}, \mathbf{d}) = -\beta \mathbf{y}^\top \log \mathbf{d} - (1 - \beta) \max(\log \mathbf{d}). \quad (6)$$

It has been shown empirically (Reed et al., 2015; Zhang et al., 2020) that BSL is noise-robust.

Generalized cross entropy (GCE): GCE is the negative Box-Cox transformation (Box and Cox, 1964) of the predicted distribution $\mathbf{d}$:

$$\text{GCE}(\mathbf{y}, \mathbf{d}) = \frac{1 - \mathbf{y}^\top (\mathbf{d}^{1-\beta})}{1 - \beta}. \quad (7)$$

GCE is equivalent to MAE when $\beta = 0$, and to CE when β approaches 1. Therefore GCE is the generalization of CE and MAE, with the sum of loss

values with respect to all classes in Eqn 1 bounded by $[\frac{K-K^\beta}{1-\beta}, \frac{K-1}{1-\beta}]$. This makes it more noise-robust than unbounded losses like CE.

Symmetric cross entropy (SCE): SCE is defined as the sum of CE and reverse cross entropy (RCE):

$$\text{SCE}(\mathbf{y}, \mathbf{d}) = -\beta \mathbf{y}^\top \log \mathbf{d} - (1-\beta)\mathbf{d}^\top \log \mathbf{y}, \tag{8}$$

with $\log 0$ defined to be a negative constant A. RCE is reduced to MAE when $A = -2$. RCE has been shown robust to label noise (Wang et al., 2019c). Similar to BSL, SCE includes a noise-robust part of RCE and non-noise-robust part of CE. [1]

Entropy regularized GCE (ER-GCE): The noise-robustness of GCE can be further improved by interpolating it with an entropy regularizer. Because both GCE and the entropy are bounded, the sum of both losses results in a noise-robust loss with tighter bounds than GCE by itself. ER-GCE is defined as

$$\text{ER-GCE}(\mathbf{y}, \mathbf{d}) = \frac{\beta(1 - \mathbf{y}^\top(\mathbf{d}^{1-\beta}))}{1-\beta} + (1-\beta)H(\mathbf{d}). \tag{9}$$

β here controls both the importance of CE in the GCE as well as the weight of the entropy term. When β approaches 1, ER-GCE still is equivalent to CE, but when β equals 0, ER-GCE is equivalent to MAE regularized by the entropy of the predicted label distribution. When β satisfies the following condition:

$$\frac{\beta(K-K^\beta)}{1-\beta} + (1-\beta)K\log K \leq \sum_{k=1}^{K} \text{ER-GCE}(\mathbf{d}, \bar{\mathbf{y}}_k) \leq \frac{\beta(K-1)}{1-\beta}, \tag{10}$$

we can show that the bounds of ER-GCE are tighter than that of GCE, indicating theoretically ER-GCE is more robust than GCE. Proofs regarding to the noise-robust properties of ER-GCE can be found in the appendix.

The noise resistance hyperparameter β in the noise-robust losses listed above controls how much noise-resistance the loss function may provide, which is a single real number tuned and kept fixed for each dataset. However, different training instances may have labels of varying quality, and we propose that noise resistance should be assessed and utilized at the instance level, explained below.

3 Instance-adaptive noise-robust training frameworks

When supervised models are used in pipelines to generate silver labels, the resulted machine-annotated dataset reflects biases and inaccuracies learned by such models, which may be caused by spurious relations between instance-level features, such as words, phrases and syntactic constructions, and labels in datasets on which these models are trained. This in turn causes some instances more likely to receive noisy silver labels than others. Ideally, each training instance should have its own noise-robustness β value, which is certainly hard to manually tune.

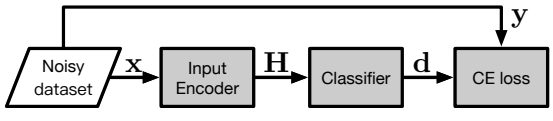
We propose that each instance should be assigned a different β with its value calculated by a label quality function $\mathcal{Q} : \{\mathbf{x}_i, y_i\} \rightarrow \beta_i, \beta_i \in (0, 1)$.[2] Motivated by the intuition that label errors in automated pipelines are correlated with difficult input features or learned biases, we model the function $\mathcal{Q}$ with a neural network, which is expected to capture the complex relationship between inputs and quality of silver labels. We propose two instance-adaptive frameworks for training classification models with noise-robust losses with instance-specific β: the first training framework jointly trains the main model and the data-quality predictor (Section 3.1) and the second one takes additional supervision of label quality and iteratively trains the main module and the data predictor (Section 3.2), shown in Figure 1b and 1c.

3.1 Joint instance-adaptive training

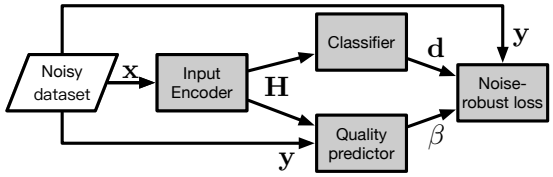
Algorithm 1 as well as Figure 1b describe the joint instance-adaptive training method. We consider a classification model to have two main components: an input encoder $\mathcal{E}$ to encode input tokens $\mathbf{x}$ into vectors $\mathbf{H}$, and a classifier $\mathcal{C}$ which makes a label prediction based on the encoded input. For example, $\mathcal{E}$ may be a neural network with an embedding layer and a multilayered BiLSTM, and $\mathcal{C}$ may be a

[1] One recent noise-robust loss derived from SCE is the normalized cross entropy with reverse cross entropy (NCE-RCE, Ma et al., 2020). Although in a similar surface form to other losses, both parts of NCE-RCE are noise-robust, and β is mostly for controling the importance of the active loss NCE, which leads to a much larger range than losses mentioned here and harder to tune. More discussion can be found in the appendix.

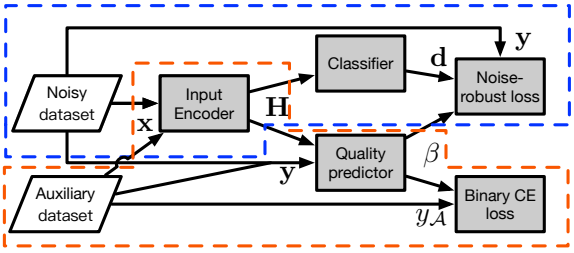
[2] The subscript i as data index is omitted in following sections for brevity. β in the following sections refers to the instance-specific β_i if not otherwise noted.



(a) Common training for classification with CE.



(b) The joint instance-adaptive noise-robust training.



(c) The iterative instance-adaptive noise-robust training. Differently colored dashed shapes indicate the models being updated in the two steps of the iterative process.

Figure 1: The instance-adaptive training frameworks with noise-robust losses.

neural network with an average pooling layer and a feedforward layer.

Given the encoded input $\mathbf{H}$ and the annotated label y, the label quality predictor $\mathcal{Q}$ learns to compare them, and predicts a β value. Although $\mathcal{Q}$ can take many complex forms depending on the prior knowledge about the relation between the inputs and the labels, two simple variants are explored in this work. The feedforward $\mathcal{Q}$ is a generic model for abstract labels such as the binary labels in paraphrase detection:

$$\hat{\beta} = \max_m(\sigma(f_{\mathcal{Q}}([\mathbf{h}_m; \mathbf{E}_{\mathcal{Q}}\mathbf{y}])), \quad (11)$$

where $\mathbf{h}_m$ is the m-th row of $\mathbf{H}$ and the encoding of m-th input token, and $\mathbf{E}_{\mathcal{Q}}$ is the embedding matrix of labels. $f_{\mathcal{Q}}$ is a neural network with feedforward layers, and σ is the sigmoid function. Intuitively, the quality predictor looks for features in the input that have the highest correlation with label quality.

For tasks where labels and inputs share direct semantic relationship such as relation extraction, the similarity-based quality predictor may be used:

$$\hat{\beta} = \max_m(\sigma(\cos(f_{\mathcal{Q}}(\mathbf{h}_m), \mathbf{E}_{\mathcal{Q}}\mathbf{y}))). \quad (12)$$

Noise-robust losses require β to be set above a threshold for good balance of robustness and fast convergence, as discussed in Section 2.1. Therefore, the final β value is lower-bounded by β_μ:

$$\beta = \hat{\beta} \times (\beta_{\text{upper}} - \beta_\mu) + \beta_\mu; \quad \beta_\mu \in (0, 1). \quad (13)$$

The common CE training scheme can be recovered when β_μ approaches $\beta_{\text{upper}} = 1$ for losses described in this work. Lower β_μ indicates higher robustness and slower convergence.

Finally, because randomly initialized models are not reliable in providing meaningful β values, the joint training framework takes advantage of a warming-up period by setting β to be 1 for a number of epochs before joint training of the quality predictor and the classification models.

Algorithm 1: Joint instance-adaptive noise-robust training

input : Training data $\mathcal{D}$, max number of epochs E, number of iterations per epoch T, noise-robust loss L
output : Trained models $\mathcal{E}^{(E)}$ $\mathcal{C}^{(E)}$, $\mathcal{Q}^{(E)}$

initialize input encoder $\mathcal{E}^{(0)}$ classifier $\mathcal{C}^{(0)}$, quality predictor $\mathcal{Q}^{(0)}$
for $e = 0$ *to* $E - 1$ **do**
  for $t = 0$ *to* $T - 1$ **do**
    $\{\mathbf{x}, \mathbf{y}\} \leftarrow$ SampleBatch($\mathcal{D}$)
    $\mathbf{H} \leftarrow \mathcal{E}(\mathbf{x}); \mathbf{d} \leftarrow \mathcal{C}(\mathbf{H})$
    $\beta \leftarrow \mathcal{Q}(\mathbf{H}, \mathbf{y})$
    $l \leftarrow L(\mathbf{y}, \mathbf{d}, \beta)$
    update all models w.r.t l
  end
end

3.2 Iterative instance-adaptive training

The iterative training framework utilizes an auxiliary dataset $\mathcal{A} = \{\mathbf{x}_j, y_j, y_{\mathcal{A}}\}_{j=1}^{J}$ to provide supervision to the quality prediction model, where $y_{\mathcal{A}} \in \{0, 1\}$ and $\mathbf{x}$ and y are from $\mathcal{D}$. Instead of correcting the original annotation which can be expensive, only manual annotation of the correctness of a label is needed. If the original label is incorrect, the auxiliary label for this training instance will be 0, otherwise it will be 1. This supervision of data quality may help the data quality predictor better capture the relationship between the input, the original label and the noise level of the instance.

Algorithm 2 and Figure 1c show how the iterative training framework is executed. $\mathcal{E}$ and $\mathcal{Q}$ are first trained by using training instances sampled from the auxiliary dataset. In the training phase of $\mathcal{E}$ and $\mathcal{C}$, the β values from $\mathcal{Q}$ are used for computing the losses, but $\mathcal{Q}$ is not updated.

Algorithm 2: Iterative instance-adaptive noise-robust training

input : Training data $\mathcal{D}$, auxiliary data $\mathcal{A}$, max number of epochs E, number of iterations per epoch $T_{\mathcal{D}}$, number of iterations per epoch $T_{\mathcal{A}}$ for auxiliary data, noise-robust loss L

output : Trained models $\mathcal{E}^{(E)}$ $\mathcal{C}^{(E)}$, $\mathcal{Q}^{(E)}$

initialize input encoder $\mathcal{E}^{(0)}$ classifier $\mathcal{C}^{(0)}$, quality predictor $\mathcal{Q}^{(0)}$
for $e = 0$ *to* $E - 1$ **do**
  for $t_{\mathcal{A}} = 0$ *to* $T_{\mathcal{A}} - 1$ **do**
    $\{\mathbf{x}, \mathbf{y}, y_{\mathcal{A}}\} \leftarrow$ SampleBatch($\mathcal{A}$)
    $\mathbf{H} \leftarrow \mathcal{E}(\mathbf{x})$
    $\beta \leftarrow \mathcal{Q}(\mathbf{H}, \mathbf{y})$
    $l_{\mathcal{A}} \leftarrow$ BinaryCrossEntropy($\beta, y_{\mathcal{A}}$)
    update $\mathcal{E}$, $\mathcal{Q}$ models w.r.t $l_{\mathcal{A}}$
  end
  for $t = 0$ *to* $T - 1$ **do**
    $\{\mathbf{x}, \mathbf{y}\} \leftarrow$ SampleBatch($\mathcal{D}$)
    $\mathbf{H} \leftarrow \mathcal{E}(\mathbf{x})$; $\mathbf{d} \leftarrow \mathcal{C}(\mathbf{H})$
    $\beta \leftarrow$ StopGradient($\mathcal{Q}(\mathbf{H}, \mathbf{y})$)
    $l \leftarrow L(\mathbf{y}, \mathbf{d}, \beta)$
    update $\mathcal{E}, \mathcal{C}$ models w.r.t l
  end
end

4 Datasets and models

Two sets of experiments with the noise-robust losses and the adaptive training frameworks are conducted to show the effectiveness of the frameworks against label noise with NLP datasets. The first set of experiments is conducted on two real noisy datasets generated by automated pipelines: a user attribute extraction dataset Getting to Know You (GTKY, Wu et al., 2020) and the English Conversational Semantic Role Labeling dataset (eCSRL, Xu et al., 2020). The GTKY dataset was created by automatically adding user attribute annotation on the PersonaChat dataset (Zhang et al., 2018). The eCSRL dataset is created by first automatically translating the hand-annotated Chinese CSRL (Xu et al., 2020) dataset to English and then aligning words and annotation (Daza and Frank, 2020) from the CSRL dataset with multilingual BERT (Devlin et al., 2019). The test sets of both datasets used in the experiments, which are subsets of the original noisy test sets, have been manually corrected by annotators.

The models used for evaluation on the noisy datasets are the user attribute extractor (Wu et al., 2020) and the biaffine semantic role labeler (Cai et al., 2018) for GTKY and eCSRL respectively. The user attribute extractor (Wu et al., 2020) has three modules: a context encoder, a predicate classifier and an entity generator. The context encoder is a BiGRU encoder, and the predicate classifier is a multi-hop memory network (Sukhbaatar et al., 2015b) which uses the all possible predicates to query the encoded input, and predict which predicates appear in the input. The entity generator is a GRU decoder with the copy mechanism where the predicate and the encoded input tokens are used as input to generate the arguments of the predicate. For example, the sentence *now I live in Florida for long.* has a predicate *live_in*, and the entities of this predicate are *I* and *Florida*. The reported score is the average F1 score of predicate prediction and entity prediction. The semantic role labeler (Cai et al., 2018) for eCSRL has a BiLSTM encoder and a biaffine scorer. The encoder first encodes the input tokens, such as a dialogue consisting of several sentences, into representations of argument candidates and predicates. The biaffine scorer compares the representation of a predicate, usually a verb in the dialogue, with representations of argument candidates through a biaffine and a feedforward layer, computing the scores of the argument candidates having an argument label. For example, for the sentence above, for the predicate *live*, *I* has the *arg0* label, but *long* has no label. The reported score is argument token F1. Auxiliary datasets are created for the noisy datasets with a budget of 12 man hours with human annotators labeling a small portion of the training instances as 0 (wrong label) or 1 (correct label), which is much easier to annotate than correcting the noisy labels.

Name	Training	Dev	Test	$\lvert\mathcal{A}\rvert$	r
GTKY	211803	24580	3000	2000	20.2%
eCSRL	24193	2999	1008	1013	17.5%
SST-2	65349	872	2000	-	-

Table 1: Relevant statistics of the datasets used in experiments. The data sizes of training, development and test sets are in number of sentences except for eCSRL which are in number of predicates. Noise rate r shows the performance of the automated pipeline evaluated against manually corrected test sets subtracted by 1, which indicates the amount of noisy labels generated by the automated pipeline.

The second set of experiments is conducted on a clean dataset with corrupted labels from the GLUE dataset (Wang et al., 2019a): SST-2 (Socher et al., 2013) for sentiment classification. Since the test set is not provided in the GLUE dataset, 2000 training instances from the training set with clean labels are arbitrarily held out as a test set. The model used in

Loss	GTKY			eCSRL			Loss Mean
	Fix	Joint	Iterative	Fix	Joint	Iterative	
CE	35.1	–	–	44.2	–	–	39.7
LS	36.3	36.7	36.4	48.4	49.1	49.2	42.7
WCE	35.2	36.4	37.8	44.3	46.5	47.3	41.2
BSL_s	36.5	38.6	39.4	49.7	49.5	50.7	44.1
BSL_h	36.5	38.4	39.3	49.2	50.1	50.9	44.1
GCE	38.6	*39.7*	39.4	*50.6*	50.7	51.2	45.0
SCE	36.3	38.1	39.1	*50.6*	*51.8*	52.1	44.7
ER-GCE	*39.1*	*39.7*	***40.5***	50.3	51.1	***52.2***	**45.5**
Framework Mean	36.9	38.2	**38.9**	49.0	49.9	**50.5**	–

Table 2: Results on the noisy datasets. For GTKY, the average F1 between predicate and entity prediction is reported. For eCSRL, argument token F1 is reported. The boldfaced numbers show the highest performance in each training framework for a dataset.**Bold numbers** indicate the best performing loss within each noise rate as well as the best mean, and *italic numbers* indicate the best performing loss within each noise rate and training framework.

this set of experiments is a BiLSTM encoder-based classifier similar to the original BiLSTM baseline model used in Wang et al. (2019a). A transformer-based model ALBERT (Lan et al., 2019) is also used in one experiment to gauge the effectiveness of proposed methods when used for finetuning. The evaluation metric is accuracy. Table 1 shows the relevant statistics of the datasets.

5 Experiments

For all experiments, a model selection procedure similar to a common use case is adopted: one β_μ is first selected from 0.1 to 0.95 in increments of 0.05, and performance of the trained models on the noisy development dataset with 3 different random seeds and the chosen β is compared. Performance of the model with the best development result is reported for the noisy datasets to simulate the common use case. Means and variances of the three models with the best performing β_μ are reported for the clean datasets for better understanding of model behavior. All other hyperparameters of the models, such as the learning rate and the batch size, are tuned with the CE loss and kept fixed.[3] The experiment conditions include six different noise-robust losses including WCE, BSL_s, BSL_h, GCE, SCE and ER-GCE, along with LS and CE as baselines. A for SCE is set to -4 following previous work (Wang et al., 2019c). They also include three training settings: **Fix** for using a fixed dataset-level β, **Joint** for using the joint noise-robust training for instance-adaptive β and **Iterative** for using the iterative framework for noise-robust training with auxiliary data. The similarity-based quality predictor is used with models trained on noisy datasets, and the feedforward one is used with models trained on corrupted datasets, as explained in Section 3.1. The warm-up period for the joint training framework is set to 5 epochs.

5.1 Noisy datasets

Results of the noisy dataset experiments, shown in Table 2, confirm the effectiveness of the instance-adaptive training frameworks. Comparing the three training frameworks, the joint training framework outperforms fixed training with a dataset-level β, showing that instance-adaptive β can help models become more noise-resistant. The iterative framework achieves the highest results, indicating distance supervision of data quality can help models further combat noisy labels. In fact, with the help from an small auxiliary set and the iterative training framework, the mean performance gains reach 4.2% and 6.5% respectively compared to CE, and 3.5% and 1.6% compared to models trained with fixed β values. Finally, comparison between noise-robust losses shows that models trained with ER-GCE are the most robust against label noise.

[3]Detailed information on the models and training procedures can be found in the appendix.

Loss	Uniform Noise												Mean
	$r = 0.2$				$r = 0.3$				$r = 0.4$				
	Fix	Joint	Iter/0.1	Iter/0.3	Fix	Joint	Iter/0.1	Iter/0.3	Fix	Joint	Iter/0.1	Iter/0.3	
CE	80.4(2.0)	–	–	–	72.6(3.4)	–	–	–	65.7(2.5)	–	–	–	
LS	83.3(0.5)	82.6(0.6)	83.8(1.9)	83.9(1.1)	75.4(2.9)	77.6(0.4)	76.1(2.2)	77.2(1.0)	65.1(3.0)	64.2(1.4)	65.4(2.1)	66.7(2.4)	75.1
WCE	80.3(1.8)	84.1(0.7)	84.0(0.7)	85.8(1.7)	72.8(3.2)	73.6(1.6)	75.1(0.8)	77.5(1.1)	66.1(0.7)	66.6(0.6)	67.6(4.1)	69.2(1.7)	75.2
BSL_s	83.0(1.1)	83.2(0.9)	84.3(0.6)	86.0(1.5)	*78.1(0.2)*	*78.6(1.9)*	79.2(0.9)	***81.7(1.9)***	65.8(1.2)	66.8(0.4)	69.8(1.8)	73.1(1.7)	77.5
BSL_h	*83.4(1.3)*	83.6(1.7)	84.8(0.2)	85.8(1.1)	75.0(5.1)	77.7(0.8)	*79.4(2.2)*	81.3(1.9)	*67.3(0.6)*	67.7(0.7)	*70.2(0.0)*	***73.6(2.9)***	77.6
GCE	83.0(1.1)	83.0(1.0)	84.6(0.7)	86.2(1.7)	77.2(1.8)	77.0(1.1)	78.1(1.8)	80.0(3.3)	66.9(0.6)	66.6(0.9)	69.2(1.5)	72.7(0.5)	77.0
SCE	83.3(1.7)	*84.4(0.8)*	*85.4(0.9)*	86.1(0.9)	77.2(1.7)	76.6(3.4)	78.6(2.1)	81.5(0.4)	66.2(2.4)	67.4(0.8)	68.8(0.7)	69.8(2.4)	77.1
ER-GCE	82.9(1.5)	83.3(1.0)	84.5(0.3)	***86.4(1.7)***	77.6(2.3)	78.4(0.9)	78.6(3.9)	***81.7(1.0)***	67.2(2.1)	*68.6(1.1)*	69.6(1.9)	73.4(2.3)	**77.7**
Mean	82.7	83.5	84.5	**85.8**	76.2	77.1	77.9	**80.2**	66.4	66.9	68.7	**70.6**	–

(a) Accuracy results on the corrupted SST-2 dataset with uniform random label noise.

Loss	Model-based Noise												Mean
	$r = 0.2$				$r = 0.3$				$r = 0.4$				
	Fix	Joint	Iter/0.1	Iter/0.3	Fix	Joint	Iter/0.1	Iter/0.3	Fix	Joint	Iter/0.1	Iter/0.3	
CE	84.5(0.7)	–	–	–	76.4(0.2)	–	–	–	59.1(4.0)	–	–	–	
LS	85.5(1.0)	85.6(1.2)	86.0(0.7)	86.1(0.5)	77.2(1.5)	78.2(0.4)	78.8(0.9)	77.8(0.5)	61.3(10.7)	65.0(8.6)	60.2(8.8)	62.1(9.1)	75.3
WCE	84.5(0.3)	85.3(0.2)	86.2(0.1)	87.1(0.2)	76.2(0.9)	77.0(2.3)	78.6(0.9)	80.2(0.2)	59.2(4.2)	60.5(5.1)	62.6(4.5)	64.8(6.8)	75.2
BSL_s	85.9(0.6)	*86.4(0.5)*	85.8(1.4)	***87.3(0.5)***	79.2(1.2)	78.2(1.5)	80.0(0.3)	81.9(0.3)	65.4(5.4)	*67.3(2.2)*	*70.1(8.7)*	***71.1(6.7)***	**78.2**
BSL_h	85.6(0.5)	85.5(0.4)	86.3(0.4)	86.9(0.4)	*79.8(0.4)*	*79.9(1.2)*	*80.6(0.2)*	81.0(0.9)	65.2(8.5)	65.6(7.7)	65.2(7.5)	68.6(2.1)	77.5
GCE	85.3(0.1)	86.2(0.7)	86.2(0.8)	***87.3(0.5)***	78.2(0.2)	77.8(0.6)	80.2(0.3)	81.8(0.4)	64.1(6.3)	65.3(4.2)	65.9(7.0)	66.9(4.1)	77.1
SCE	85.8(0.7)	85.6(0.3)	86.2(0.3)	86.7(1.1)	78.1(0.9)	79.1(1.0)	79.1(0.6)	81.7(0.6)	61.0(8.3)	66.6(4.5)	68.1(2.1)	70.1(5.2)	77.3
ER-GCE	*86.2(0.3)*	85.9(0.6)	*86.5(1.3)*	87.2(1.8)	78.4(0.2)	78.9(0.8)	*80.6(0.7)*	***82.0(1.0)***	*65.5(7.2)*	66.6(6.5)	69.7(4.2)	***71.1(7.6)***	**78.2**
Mean	85.6	85.8	86.2	**87.0**	78.1	78.5	79.7	**80.9**	63.1	65.3	65.9	**67.8**	–

(b) Accuracy results on the corrupted SST-2 dataset with model-based label noise.

Table 3: Accuracy results on the corrupted SST-2 datasets. Means and standard deviations are reported for each experiment condition using the best β_μ on the development set, and the means of all means across different losses for each noise and training condition, and all means across different noise and training conditions for each loss, are reported in the final row and column for marginalized comparisons. Noise rates $r \in \{0.2, 0.3, 0.4\}$ are the rates with which the clean labels are corrupted. Iter/0.1 and Iter/0.3 mean experiments with the iterative training framework with an auxiliary dataset containing 10% and 30% of the training instances respectively. **Bold numbers** and *italic numbers* have the same meaning as Table 2.

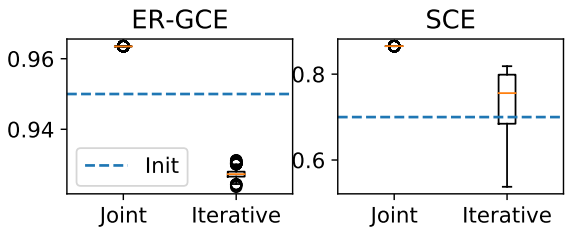
5.2 Clean datasets with corrupted labels

The relationship between noise rates, training frameworks and noise-robust losses are further explored with the clean SST-2 dataset where the noise rate and the size of the auxiliary dataset can be easily manipulated. In these experiments, we randomly corrupt the original labels at a noise rate $r \in \{0.2, 0.3, 0.4\}$[4] and construct an auxiliary dataset with 10% or 30% of the corrupted training set. The label corruption is done through two different ways: the uniform noisy datasets are created by randomly corrupting labels to reach a noise rate, and the model-based noisy datasets are created with a five-fold cross-corruption process: an ALBERT-based classifier is trained on four-fifths of the clean training set, and labels of instances in the held-out one-fifth set in which the trained model has lowest confidence are corrupted to reach a noise rate.[5]

Table 3 show the model performances under various experiment conditions. First, there are significant differences between the best performing models and the baseline models trained with CE, which can reach 9.7%. Similarly, the performance difference between the iterative models and the fix models can reach 5.4%. These significant performance gains showcase again the value of the proposed frameworks for training with noisy datasets. Also, experiment data on different noise-robust

[4]Previous work with vision datasets often includes noise rates up to 0.9. We consider this to be extremely impractical if the trained models are to be useful for some task.

[5]Further detail about the model-based noising process can be found in the appendix.



(a) β values after training with joint and iterative frameworks with ER-GCE and SCE on eCSRL.

Corrupted %	Fix	Joint	Iter/0.1	Iter/0.3
0%	80.0(0.0)	80.6(0.1)	84.4(6.9)	87.5(6.6)
50%	80.0(0.0)	80.5(0.1)	82.8(4.4)	83.4(5.1)
100%	80.0(0.0)	80.2(0.1)	80.8(2.9)	81.6(3.9)

(b) Distribution of βs for test instances with different test label corruption rate with ER-GCE on SST-2.

Figure 2: β values reflect data quality.

losses indicates ER-GCE loss to be the most noise-robust among the losses explored. Second, the performance trend between these training frameworks is similar to what has been shown with noisy datasets: the joint training framework outperforms the fixed training framework consistently, and the iterative training framework provides a further boost to model performance compared to joint and fixed frameworks.

5.3 Model analysis

β values reflect instance quality: Figure 2a shows how different training frameworks influence the distribution of β for the best-performing $\beta_m u$ values on eCSRL, which are 0.9 for ER-GCE and 0.4 for SCE. The dashed line indicates the initial value at $\beta_\mu + (1-\beta_\mu)/2$. The final instance-specific β values trained with the joint framework tend to concentrate around a value different from the original β_μ and the initial value, showing the adaptive nature of the training framework. The small amount of auxiliary quality data is able to increase the variance of the individual β values, indicating that the quality model has learned to assign different β values to different training instances according to their label quality.

This can be seen in Table 2b, which shows means and standard deviations of predicted β values for test instances of SST-2 when a portion of the test labels is also corrupted. The models are trained with ER-GCE with 30% model-based noise in the training set. Because the test instances are not seen in training, the predicted β values represent assessment of data quality by the model. The beta values from models trained iteratively are much higher when no label is corrupted compared to when all labels are corrupted, indicating that the quality predictors are able to make generalizable judgments about data quality.

r	Fix	Joint	Iter/0.1	Iter/0.3
0.2	88.2↑ 2.0	88.3↑ 2.4	88.9↑ 2.4	90.3↑ 3.1
0.3	87.1↑ 8.7	87.4↑ 8.5	88.8↑ 8.2	88.9↑ 6.9
0.4	83.7↑ 18.2	84.0↑ 17.4	85.0↑ 15.3	85.3↑ 14.2

Table 4: Performance of the classifiers with finetuned ALBERT models on SST-2 dataset with model-based corruption with ER-GCE. ↑ indicates the performance difference between an ALBERT model and BiLSTM model under the same condition.

Finetuning benefits from noise-robust training: Finally, the BiLSTM encoder is replaced by a pre-trained ALBERT (Lan et al., 2019) base model for evaluating proposed methods in the finetuning framework. Table 4 shows the average accuracy values in various experiment conditions as well as the performance difference compared to BiLSTM models in Table 3. Results show that the proposed methods also work with the popular finetuning paradigm, achieving better results in all experiment conditions and further weakening the harmful influence of noisy labels.

6 Related work

There have been many different approaches to address the noisy label problem. One such approach relies on knowledge of clean labels (Xiao et al., 2015; Li et al., 2017; Lee et al., 2018), while another tries to estimate the label-dependent (Natarajan et al., 2013; Patrini et al., 2017) or annotator-dependent (Khetan et al., 2018) noise distributions, many with neural network layers (Sukhbaatar et al., 2015a; Bekker and Goldberger, 2016; Goldberger and Ben-Reuven, 2017). Such methods have seen some application in natural language processing (Hedderich and Klakow, 2018; Lange et al., 2019; Wang et al., 2019b). Different training strategies have also been proposed to increase the robustness (Huang et al., 2020), many of which require training of auxiliary networks to reweight samples (Jiang et al., 2018; Han et al., 2018; Wang et al., 2019b). Complementary labels (Ishida et al., 2017; Yu et al., 2018) are also used for negative learning for robustness (Kim et al., 2019; Shu et al.,

2019). Regularization techniques such as drop-out (Srivastava et al., 2014; Li et al., 2020) also show positive results in combating noisy labels. Hu et al. (2020) proposed adding auxiliary variables into normal loss functions for regularization, which act as instance-specific priors over the predicted distributions to ease the training difficulty when labels are noisy. Conceptually similar to the joint training with instance-adaptive βs proposed in this work, the regularization method in (Hu et al., 2020) may be complementary to noise-robust losses explored in this work, because noise-robust losses may enjoy further improvement when combined with trainable instance-specific priors.

Noise-robust losses are another way to counter label noise (Beigman and Beigman Klebanov, 2009). The noise-robustness of some losses, such as MAE, was shown theoretically in Ghosh et al. (2017). New noise-robust losses have also been proposed where some of the losses have a passive component attached to CE for noise-robustness (Reed et al., 2015; Wang et al., 2019c), which was used by Zhang et al. (2020) for reading comprehension with noisy data. Others behave like a mixture of MAE and CE (Zhang and Sabuncu, 2018). Other methods, such as normalization (Ma et al., 2020) of losses and use of determinant-based mutual information (Xu et al., 2019) as a finetuning loss have also shown to be robust to noise.

7 Conclusion

This work focuses on combating noisy labels in NLP datasets by means of adaptive training with noise-robust losses. Two novel instance-adaptive training frameworks are proposed and investigated along with several noise-robust losses including a new ER-GCE loss. Experiments on different datasets show the effectiveness of the approach: the adaptive training frameworks help models achieve the best performance on noisy datasets, and the ER-GCE shows great noise-robustness among the previously proposed losses.

References

Eyal Beigman and Beata Beigman Klebanov. 2009. Learning with annotation noise. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 280–287, Suntec, Singapore. Association for Computational Linguistics.

Alan Joseph Bekker and Jacob Goldberger. 2016. Training deep neural-networks based on unreliable labels. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2682–2686. IEEE.

George EP Box and David R Cox. 1964. An analysis of transformations. *Journal of the Royal Statistical Society: Series B (Methodological)*, 26(2):211–243.

Jiaxun Cai, Shexia He, Zuchao Li, and Hai Zhao. 2018. A full end-to-end semantic role labeler, syntactic-agnostic over syntactic-aware? In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2753–2765, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Angel Daza and Anette Frank. 2020. X-SRL: A parallel cross-lingual semantic role labeling dataset. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3904–3914, Online. Association for Computational Linguistics.

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. Association for Computational Linguistics.

Marthinus Du Plessis, Gang Niu, and Masashi Sugiyama. 2014. Analysis of learning from positive and unlabeled data. In *Advances in neural information processing systems*, 27, pages 703–711.

Aritra Ghosh, Himanshu Kumar, and P S Sastry. 2017. Robust loss functions under label noise for deep neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 31.

Jacob Goldberger and Ehud Ben-Reuven. 2017. Training deep neural networks using a noise adaptation layer. In *Proc. 5^{th} International Conference on Learning Representations*, Palais des Congrès Neptune, Toulon, France.

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. In *Advances in Neural Information Processing Systems*,

volume 31, pages 8527–8537. Curran Associates, Inc.

Kazuma Hashimoto, Caiming Xiong, Yoshimasa Tsuruoka, and Richard Socher. 2017. A joint many-task model: Growing a neural network for multiple NLP tasks. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1923–1933, Copenhagen, Denmark. Association for Computational Linguistics.

Michael A Hedderich and Dietrich Klakow. 2018. Training a neural network in a Low-Resource setting on automatically annotated noisy data. In *Proceedings of the Workshop on Deep Learning Approaches for Low-Resource NLP*, pages 12–18, Melbourne. Association for Computational Linguistics.

Wei Hu, Zhiyuan Li, and Dingli Yu. 2020. Simple and effective regularization methods for training on noisily labeled data with generalization guarantee. In *Proceedings of the International Conference on Learning Representations*.

Lang Huang, Chao Zhang, and Hongyang Zhang. 2020. Self-Adaptive training: beyond empirical risk minimization. In *Advances in Neural Information Processing Systems*, 33.

Takashi Ishida, Gang Niu, Weihua Hu, and Masashi Sugiyama. 2017. Learning from complementary labels. In *Advances in neural information processing systems*, pages 5639–5649.

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.

Mandar Joshi, Eunsol Choi, Daniel 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, Vancouver, Canada. Association for Computational Linguistics.

Ashish Khetan, Zachary C Lipton, and Animashree Anandkumar. 2018. Learning from noisy singly-labeled data. In *International Conference on Learning Representations*.

Youngdong Kim, Junho Yim, Juseung Yun, and Junmo Kim. 2019. NLNL: Negative learning for noisy labels. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE.

Alex Krizhevsky. 2009. Learning multiple layers of features from tiny images. Technical report, University of Toronto.

Wuwei Lan, Siyu Qiu, Hua He, and Wei Xu. 2017. A continuously growing dataset of sentential paraphrases. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1224–1234, Copenhagen, Denmark. Association for Computational Linguistics.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. ALBERT: A lite BERT for self-supervised learning of language representations.

Lukas Lange, Michael A. Hedderich, and Dietrich Klakow. 2019. Feature-dependent confusion matrices for low-resource NER labeling with noisy labels. 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 3554–3559, Hong Kong, China. Association for Computational Linguistics.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Kuang-Huei Lee, Xiaodong He, Lei Zhang, and Linjun Yang. 2018. CleanNet: Transfer learning for scalable image classifier training with label noise. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE.

Mingchen Li, Mahdi Soltanolkotabi, and Samet Oymak. 2020. Gradient descent with early stopping is provably robust to label noise for overparameterized neural networks. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 4313–4324. PMLR.

Yuncheng Li, Jianchao Yang, Yale Song, Liangliang Cao, Jiebo Luo, and Li-Jia Li. 2017. Learning from noisy labels with distillation. In *2017 IEEE International Conference on Computer Vision (ICCV)*. IEEE.

Xingjun Ma, Hanxun Huang, Yisen Wang, Simone Romano, Sarah Erfani, and James Bailey. 2020. Normalized loss functions for deep learning with noisy labels. In *International Conference on Machine Learning*, pages 6543–6553.

Nagarajan Natarajan, Inderjit S Dhillon, Pradeep K Ravikumar, and Ambuj Tewari. 2013. Learning with noisy labels. In *Advances in Neural Information Processing Systems*, volume 26, pages 1196–1204. Curran Associates, Inc.

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.

Debjit Paul, Mittul Singh, Michael A. Hedderich, and Dietrich Klakow. 2019. Handling noisy labels for

robustly learning from self-training data for low-resource sequence labeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 29–34, Minneapolis, Minnesota. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Scott E Reed, Honglak Lee, Dragomir Anguelov, Christian Szegedy, Dumitru Erhan, and Andrew Rabinovich. 2015. Training deep neural networks on noisy labels with bootstrapping.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Jun Shu, Qi Xie, Lixuan Yi, Qian Zhao, Sanping Zhou, Zongben Xu, and Deyu Meng. 2019. Meta-Weight-Net: Learning an explicit mapping for sample weighting. In H Wallach, H Larochelle, A Beygelzimer, F d'Alché-Buc, E Fox, and R Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 1919–1930.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Diana Sousa, Andre Lamurias, and Francisco M. Couto. 2019. A silver standard corpus of human phenotype-gene relations. 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 1487–1492, Minneapolis, Minnesota. Association for Computational Linguistics.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958.

Sainbayar Sukhbaatar, Joan Bruna Estrach, Manohar Paluri, Lubomir Bourdev, and Robert Fergus. 2015a. Training convolutional networks with noisy labels. In *3rd International Conference on Learning Representations, ICLR 2015*.

Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. 2015b. End-to-end memory networks. *Advances in Neural Information Processing Systems*, 2015:2440–2448.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the Inception Architecture for Computer Vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2019a. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *7th International Conference on Learning Representations, ICLR 2019*.

Hao Wang, Bing Liu, Chaozhuo Li, Yan Yang, and Tianrui Li. 2019b. Learning with noisy labels for sentence-level sentiment classification. 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 6286–6292, Hong Kong, China. Association for Computational Linguistics.

Yisen Wang, Xingjun Ma, Zaiyi Chen, Yuan Luo, Jinfeng Yi, and James Bailey. 2019c. Symmetric cross entropy for robust learning with noisy labels. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE.

Chien-Sheng Wu, Andrea Madotto, Zhaojiang Lin, Peng Xu, and Pascale Fung. 2020. Getting to know you: User attribute extraction from dialogues. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 581–589, Marseille, France. European Language Resources Association.

Tong Xiao, Tian Xia, Yi Yang, Chang Huang, and Xiaogang Wang. 2015. Learning from massive noisy labeled data for image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2691–2699.

Kun Xu, Haochen Tan, Linfeng Song, Han Wu, Haisong Zhang, Linqi Song, and Dong Yu. 2020. Semantic Role Labeling Guided Multi-turn Dialogue ReWriter. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6632–6639, Online. Association for Computational Linguistics.

Yilun Xu, Peng Cao, Yuqing Kong, and Yizhou Wang. 2019. L_DMI: A novel information-theoretic loss function for training deep nets robust to label noise. In *Advances in Neural Information Processing Systems*, volume 32, pages 6225–6236. Curran Associates, Inc.

Xiyu Yu, Tongliang Liu, Mingming Gong, and Dacheng Tao. 2018. Learning with biased complementary labels. In *Proceedings of the European*

Conference on Computer Vision (ECCV), pages 68–83.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. Personalizing dialogue agents: I have a dog, do you have pets too? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2204–2213, Melbourne, Australia. Association for Computational Linguistics.

Xuemiao Zhang, Kun Zhou, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Junfei Liu. 2020. Learn with noisy data via unsupervised loss correction for weakly supervised reading comprehension. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2624–2634, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Zhilu Zhang and Mert Sabuncu. 2018. Generalized cross entropy loss for training deep neural networks with noisy labels. In *Advances in Neural Information Processing Systems 31*, pages 8778–8788. Curran Associates, Inc.

A Discussion about Normalized cross entropy with RCE (NCE-RCE)

Normalization of CE can transform the CE loss to a noise-robust loss because the probabilities are bounded unlike the logarithms of probabilities. NCE uses the normalized negative log-probabilities as loss values. Ma et al. (2020) further suggests combining an active loss such as NCE where the probability of the annotated label is explicitly maximized, with a passive loss such as RCE where the probability of at least one unannotated label is explicitly minimized. The definition of NCE-RCE is the weighted sum of both losses:

$$\text{NCE-RCE}(\mathbf{y}, \mathbf{d}) = \frac{-\beta \mathbf{y}^\top \log \mathbf{d}}{-\mathbf{1}^\top \log \mathbf{d}} - (1-\beta)\mathbf{d}^\top \log \mathbf{y}, \tag{14}$$

and $\log 0$ is defined to be a negative constant A. Because both component losses are noise-robust, β here tunes how much active learning is in the interpolation, which may be correlated to dataset complexity (Ma et al., 2020). The β values for NCE-RCE in the previous work Ma et al. (2020) usually include $\{0.001, 0.01, 0.99, 0.999\}$. This indicates different function of the hyperparameter β compared to noise-robust losses examined in the paper, which is usually how noisy an instance is. Preliminary experiments also support this observation, where NCE-RCE tends to underfit and perform poorly with β values close to the true error rate.

B Lower bound of sum of losses with respect to all classes

Typically loss functions penalize prediction distributions which are further away from the gold labels than ones closer at least equally or more:

$$L(\mathbf{d}_k - \Delta\mathbf{d}_k, k) - L(\mathbf{d}_k, k) \geq L(\mathbf{d}'_k - \Delta\mathbf{d}_k, k) - L(\mathbf{d}'_k, k) \tag{15}$$

if $0 \leq \Delta\mathbf{d}_k \leq \mathbf{d}_k \leq \mathbf{d}'_k$.[6] The lower bound of $\sum_{k=1}^{K} L(\mathbf{d}, k)$ is $\mathbf{d}$ at uniform if Eqn 15 is satisfied.

Proof. Suppose $\tilde{\mathbf{d}}$ is a vectorial representation of a uniform categorical distribution where $\tilde{\mathbf{d}}_k = \frac{1}{K}$ for $k \in \{1, ..., K\}$. If $\Delta\mathbf{d}_{k_1}$ is moved from $\mathbf{d}_{k_1}$ to $\mathbf{d}_{k_2}$ for $k_1, k_2 \in \{1, ..., K\}, k_1 \neq k_2$ where $0 \leq \Delta\mathbf{d}_{k_1}$, because $\tilde{\mathbf{d}}_{k_2} + \Delta\mathbf{d}_{k_1} \geq \tilde{\mathbf{d}}_{k_2} = \tilde{\mathbf{d}}_{k_1} \geq \tilde{\mathbf{d}}_{k_1} - \Delta\mathbf{d}_{k_1}$, according to Eqn 15, we have

$$L(\tilde{\mathbf{d}}_{k_1}, k_1) - L(\tilde{\mathbf{d}}_{k_1} - \Delta\mathbf{d}_{k_1}, k_1) \leq L(\tilde{\mathbf{d}}_{k_2} + \Delta\mathbf{d}_{k_1}, k_2) - L(\tilde{\mathbf{d}}_{k_2}, k_2),$$

then the change in the sum of losses with respect to all classes is

$$\begin{aligned}
&\sum_{k\in\{1,...,K\}} L(\tilde{\mathbf{d}}, k) - \sum_{k\in\{1,...,K\}\setminus\{k_1,k_2\}} L(\tilde{\mathbf{d}}, k) - \\
&\quad L(\tilde{\mathbf{d}}_{k_1} - \Delta\mathbf{d}_{k_1}, k_1) - L(\tilde{\mathbf{d}}_{k_2} + \Delta\mathbf{d}_{k_1}, k_2) \\
&= L(\tilde{\mathbf{d}}_{k_1}, k_1) - L(\tilde{\mathbf{d}}_{k_1} - \Delta\mathbf{d}_{k_1}, k_1) \\
&\quad + L(\tilde{\mathbf{d}}_{k_2}, k_2) - L(\tilde{\mathbf{d}}_{k_2} + \Delta\mathbf{d}_{k_1}, k_2) \\
&\leq 0.
\end{aligned}$$

This shows that any change to the uniform vector causes the sum to increase, thus proving the lower bound can be found at the uniform vector. □

In the case of ER-GCE, when $\beta \in [0, 1)$, for the GCE part of the loss, the sum of the loss with respect to all classes is bounded by:

$$\frac{K - K^\beta}{1-\beta} \leq \sum_{k=1}^{K} \frac{1 - \mathbf{d}_k^{1-\beta}}{1-\beta} \leq \frac{K-1}{1-\beta}, \tag{16}$$

where $\mathbf{d}_k$ is the k-th element of the prediction vector (Zhang and Sabuncu, 2018). The lower bound of GCE is at $\mathbf{d}$ being a uniform categorical distribution, and upper bound is at $\mathbf{d}$ being a one-hot categorical distribution. However, the entropy part of ER-GCE has a lower bound when $\mathbf{d}$ being one-hot, and a upper bound when $\mathbf{d}$ being uniform:

$$0 \leq \sum_{k=1}^{K} H(\mathbf{d}) \leq K \log K. \tag{17}$$

Therefore, for ER-GCE loss where the GCE part and the entropy part are summed up, β needs to satisfy the following condition for good learning behavior:

$$\frac{\beta(K - K^\beta)}{1-\beta} + (1-\beta)K\log K \leq \sum_{k=1}^{K} \text{ER-GCE}(\mathbf{d}, \bar{\mathbf{y}}_k) \leq \frac{\beta(K-1)}{1-\beta} \tag{18}$$

with the lower bound being the sum of losses with regard to all classes for ER-GCE at uniform, and the upper bound being the sum at one-hot.

[6] Intuitively, this means the loss increases the same or more in absolute value when the prediction moves further away from the true label than moving closer to the true label.

C Theorem 1: Noise-robustness of ER-GCE under uniform noise

Under uniform noise with $\eta \leq 1 - \frac{1}{K}$ which is the probability of the true label $\hat{y}$ being corrupted to the observed label y,

$$0 \leq R^{\eta}_{L_\beta}(f^*) - R^{\eta}_{L_\beta}(\hat{f}) \leq A$$

where $A = \frac{\eta(\psi-\phi)}{K-1} \geq 0$, the bounds of ER-GCE in Eqn 18 are $[\phi, \psi]$, f^* is the global minimizer of risk $R_{L_\beta}(f)$ and $\hat{f}$ is the global minimizer of the risk $R^{\eta}_{L_\beta}(f)$.

Proof. For the model f, the empirical risk of the model with a dataset $\mathcal{D}$ is

$$R_{L_\beta}(f) = \mathbb{E}_{\{\mathbf{x},y\}\sim\mathcal{D}}[L_\beta(f(\mathbf{x}), y)].$$

When the noise is uniform with noise rate η where $\eta_{jk} = 1 - \eta$ for $j = k$ and $\eta_{jk} = \frac{\eta}{k-1}$ for $j \neq k$, we have:

$$\begin{aligned}
R^{\eta}_{L_\beta}(f) &= \mathbb{E}_{\mathbf{x}}\mathbb{E}_{\hat{y}|\mathbf{x}}\mathbb{E}_{y|\hat{y},x}[L_\beta(f(\mathbf{x}, y))] \\
&= \mathbb{E}_{\mathbf{x}}\mathbb{E}_{\hat{y}|\mathbf{x}}[(1-\eta)L_\beta(f(\mathbf{x}, y)) + \\
&\quad \frac{\eta}{K-1}\sum_{k\neq y} L_\beta(f(\mathbf{x}), k)] \\
&= (1 - \frac{\eta K}{K-1})R_{L_\beta}(f) + \\
&\quad \frac{\eta}{K-1}\mathbb{E}_{\mathbf{x}}\mathbb{E}_{\hat{y}|\mathbf{x}}[\sum_{k=1}^{K} L_\beta(f(\mathbf{x}, k)].
\end{aligned}$$

Let the bounds of ER-GCE in Eqn 18 be $[\phi, \psi]$, the bounds of the risk with noise can be written as:

$$\begin{aligned}
(1 - \frac{\eta K}{K-1})R_{L_\beta}(f) + \frac{\eta\phi}{K-1} &\leq R^{\eta}_{L_\beta} \\
&\leq (1 - \frac{\eta K}{K-1})R_{L_\beta}(f)\frac{\eta\psi}{(K-1)}.
\end{aligned}$$

Let f^* be the global minimizer of risk $R_{L_\beta}(f)$ and $\hat{f}$ be the global minimizer of the risk $R^{\eta}_{L_\beta}(f)$. When $R^{\eta}_{L_\beta}(f^*) - R^{\eta}_{L_\beta}(\hat{f}) = 0$, the loss L at β is completely noise-robust, meaning the optimal model trained with noisy or clean data has no different in risk. For $\hat{f}$ and ER-GCE loss,

$$\begin{aligned}
R^{\eta}_{L_\beta}(f^*) - R^{\eta}_{L_\beta}(\hat{f}) &\leq A + \\
(1 - \frac{\eta K}{K-1})(R_{L_\beta}(f^*) - R_{L_\beta}(\hat{f})) &\leq A,
\end{aligned}$$

where $A = \frac{\eta(\psi-\phi)}{K-1} \geq 0$. Since f^* is the minimizer of $R_{L_\beta}(f)$ and $\hat{f}$ is the minimizer of $R^{\eta}_{L_\beta}(f)$, $R^{\eta}_{L_\beta}(f^*) - R^{\eta}_{L_\beta}(\hat{f}) \geq 0$. As β decreases and ψ approaches ϕ, A approaches 0, making the loss more tolerant to noise. □

D Theorem 2: Noise robustness of ER-GCE under class-dependent noise

Under class-dependent noise when $\eta_{jk} < (1 - \eta_j), \forall j \neq k, \forall j, k \in 1, ..., K$, where $\eta_{jk} = p(y = k|\hat{y} = j), \forall j \neq k$, and $(1 - \eta_j) = p(y = j|\hat{y} = j)$, if we have $R_{L_\beta}(f^*) = 0$, then

$$0 \leq R^{\eta}_{L_\beta}(f^*) - R^{\eta}_{L_\beta}(\hat{f}) \leq B,$$

where $B = (\psi - \phi)\mathbb{E}_{\{\mathbf{x},y\}\sim\mathcal{D}}[1 - \eta_y] \geq 0$. f^* is the global minimizer of risk $R_{L_\beta}(f)$ and $\hat{f}$ is the global minimizer of the risk $R^{\eta}_{L_\beta}(f)$.

Proof. Similar to Theorem 1, under this noise model, we have:

$$\begin{aligned}
&R^{\eta}_{L_\beta}(f) \\
&= \mathbb{E}_{\{\mathbf{x},y\}\sim\mathcal{D}}[(1-\eta_y)L_\beta(f(\mathbf{x}), y)] \\
&\quad + \mathbb{E}_{\{\mathbf{x},y\}\sim\mathcal{D}}[\sum_{k\neq y}\eta_{yk}L_\beta(f(\mathbf{x}), k)] \\
&\leq \mathbb{E}_{\{\mathbf{x},y\}\sim\mathcal{D}}[(1-\eta_y)(\psi - \sum_{k\neq y} L_\beta(f(\mathbf{x}), k))] \\
&\quad + \mathbb{E}_{\{\mathbf{x},y\}\sim\mathcal{D}}[\sum_{k\neq y}\eta_{yk}L_\beta(f(\mathbf{x}), k)] \\
&= \psi\mathbb{E}_{\{\mathbf{x},y\}\sim\mathcal{D}}[1-\eta_y] \\
&- \mathbb{E}_{\{\mathbf{x},y\}\sim\mathcal{D}}[\sum_{k\neq y}(1-\eta_y-\eta_{yk})(L_\beta(f(\mathbf{x}), k))].
\end{aligned}$$

Similarly, we also have:

$$\begin{aligned}
&R^{\eta}_{L_\beta}(f) \geq \phi\mathbb{E}_{\{\mathbf{x},y\}\sim\mathcal{D}}[1-\eta_y] \\
&- \mathbb{E}_{\{\mathbf{x},y\}\sim\mathcal{D}}[\sum_{k\neq y}(1-\eta_y-\eta_{yk})(L_\beta(f(\mathbf{x}), k))].
\end{aligned}$$

Therefore,

$$\begin{aligned}
R^{\eta}_{L_\beta}(f^*) - R^{\eta}_{L_\beta}(\hat{f}) &\leq (\psi-\phi)\mathbb{E}_{\{\mathbf{x},y\}\sim\mathcal{D}}[1-\eta_y] \\
&+ \mathbb{E}_{\{\mathbf{x},y\}\sim\mathcal{D}}[\sum_{k\neq y}(1-\eta_y-\eta_{yk})(L_\beta(\hat{f}(\mathbf{x}), k) \\
&\quad - L_\beta(f^*(\mathbf{x}), k))]
\end{aligned}$$

Because f^* is the global minimizer with empirical risk being 0 under the assumption, $f^*(\mathbf{x})_k = 1$ when $k = y$ and $f^*(\mathbf{x})_k = 0$ when $k \neq y$. Therefore, the ER-GCE loss $L_\beta(f^*(\mathbf{x}), k) = \frac{\beta}{1-\beta}, \forall k \neq y$. Because $1 - \eta_y - \eta_{yk} > 0$ under our assumption,

and $L_\beta(\hat{f}(\mathbf{x}), k) - L_\beta(f^*(\mathbf{x}), k) \leq 0$ because of Lemma 1, therefore the right-hand side can be maximized when $L_\beta(\hat{f}(\mathbf{x}), k) = L_\beta(f^*(\mathbf{x}), k)$. With this, we have:

$$R^\eta_{L_\beta}(f^*) - R^\eta_{L_\beta}(\hat{f}) \leq (\psi - \phi)\mathbb{E}_{\{\mathbf{x},y\}\sim\mathcal{D}}[1 - \eta_y]$$

Since f^* is the minimizer of $R_{L_\beta}(f)$ and $\hat{f}$ is the minimizer of $R^\eta_{L_\beta}(f)$, $R^\eta_{L_\beta}(f^*) - R^\eta_{L_\beta}(\hat{f}) \geq 0$. Similar to Theorem 1, as ψ approaches ϕ, the bounds get closer and closer, making the loss more tolerant to noise. □

E Lemma 1: Tighter bounds of ER-GCE

ER-GCE has tighter bounds than GCE with a given $\beta \in [0, 1)$ when Eqn 18 is satisfied.

Proof. We can compare the lower and upper bounds of ER-GCE and GCE:

$$\begin{aligned}
&\frac{\beta(K-1)}{1-\beta} - \frac{\beta(K-K^\beta)}{1-\beta} - (1-\beta)K\log K \\
&\quad - \frac{K-1}{1-\beta} + \frac{K-K^\beta}{1-\beta} \\
&= \frac{(\beta-1)(K-1)}{1-\beta} + \frac{(1-\beta)(K-K^\beta)}{1-\beta} \\
&\quad - (1-\beta)K\log K \\
&= 1 - K^\beta - (1-\beta)K\log K
\end{aligned}$$

Since $K > 1$, $K^\beta > 1$, therefore $1 - K^\beta < 0$ and $-(1-\beta)K\log K < 0$. Since the range difference between the bounds is negative, ER-GCE has a smaller range or tighter bounds than GCE for a given β. □

F Model structures and hyperparameters

F.1 GTKY

The model proposed by Wu et al. (2020) serves as our baseline model for the GTKY dataset trained with different noise-robust losses. There are three modules in the model: a context encoder that consumes the given word sequence $w_1, w_2, \ldots, w_N$, a relation classifier that predicts each associated relation (e.g. r), and an entity generator that generates the subject s and object o strings for a given relation r.

Context Encoder The context encoder encodes input tokens with embeddings, which are then consumed by a bi-directional GRU (Cho et al., 2014) layer. The resulting hidden states are $\mathbf{H} = [\mathbf{h}_1, \ldots, \mathbf{h}_N]$, where $\mathbf{h}_t = [\overleftarrow{\mathbf{h}}_t; \overrightarrow{\mathbf{h}}_t]$, the concatenation of forward and backward hidden states.

Relation Classifier This classifier is a I-hop (I is set to 3 following Wu et al., 2020) end-to-end memory network (Sukhbaatar et al., 2015b), which takes the hidden states from the context encoder as its input queries. The memory $\mathbf{M}^i \in \mathbb{R}^{K\times D}$ at each hop i are trainable parameters that contain the representations for all candidate relations, where K and D indicate the number of candidate relations and memory depth, respectively. The memory representation for each relation (e.g. *live_in*) is initialized by averaging the embeddings of its words (*live* and *in*). At each hop i, the attention scores between the query $\mathbf{q}^i \in \mathbb{R}^D$ and the corresponding memory are computed as:

$$\boldsymbol{\alpha}^i = softmax(\mathbf{M}^i\mathbf{q}^i).$$

Here $\boldsymbol{\alpha}^i$ is a distribution over all relations, showing model confidence at layer i on what relations are mentioned in the given text. The memory update is computed as the weighted sum of the current memory matrix:

$$\mathbf{o}^k = \boldsymbol{\alpha}^k\mathbf{M}^{k+1}$$

The first query $\mathbf{q}^1$ is initialized as $\mathbf{h}_N$, and the query at each step i is updated by:

$$\mathbf{q}^{k+1} = \mathbf{q}^k + \mathbf{o}^k$$

In the final layer, we apply a sigmoid function to trigger relations independently such that we can extract all possible relations from the given text:

$$p_j = \sigma(\mathbf{m}_j^{K+1}\mathbf{q}^{K+1}),$$

where $\mathbf{m}_j^{K+1} \in \mathbf{M}^{K+1}$ corresponds to the j-th relation.

Entity Generator Given each predicted relation r, the entity generator aims to generates the corresponding subject s and object o phrases to complete the final user attribute (s, r, o). The entity generator generates the word sequence $(\tilde{w}_1, \ldots, \tilde{w}_M)$ of concatenated subject and object, where the boundary is represented by a semicolon. For instance, the corresponding word sequence for triplet "(My

son, misc_attr, shy)" is "my son ; shy". The model is a GRU decoder (Cho et al., 2014) with a copy mechanism (See et al., 2017) for easier generation of the words that also appear in the inputs. The final distribution over the vocabulary at timestep t is calculated as

$$P_t^{\text{final}} = P^{\text{gen}} P_t^{\text{vocab}} + (1 - P^{\text{gen}}) P_t^{\text{source}},$$

where $P_t^{\text{vocab}} = \text{softmax}(\mathbf{W}\mathbf{h}_t^{\text{dec}})$ is a predicted distribution over the whole vocabulary, and $\mathbf{h}_t^{\text{dec}}$ is the hidden state of the GRU. $P_t^{\text{source}} = \text{softmax}(\mathbf{H}\mathbf{h}_t^{\text{dec}})$ is a distribution over the input tokens, and finally P^{gen} controls how they mix:

$$P^{\text{gen}} = \sigma(\mathbf{W}'[\mathbf{h}_t^{\text{dec}}; \tilde{w}_{t-1}; v_c]),$$

where $v_c = \text{diag}(P_t^{\text{source}})\ \mathbf{H}$, and $\mathbf{W}, \mathbf{W}'$ are model parameters.

Hyperparameters The hidden state sizes for all modules are set to 400, with the input embeddings intialized with Glove (Pennington et al., 2014) and character embeddings (Hashimoto et al., 2017). The batch size is set to 32. The models are optimized with Adam with learning rate set to 1×10^{-3}. Dropout layers with dropout rate 0.6 are applied to all layer transitions. Model performance is evaluated on the development set every epoch, and training is stopped whenever there is no observed improvement in development F1 score in 6 evaluations starting at the 10-th epoch.

F.2 eCSRL

The model proposed by Cai et al. (2018) serves as our baseline model trained with different noise-robust losses. There are two modules in the model: a context encoder that takes the given word sequence $w_1, w_2, \ldots, w_N$, and an biaffine role scorer which predicts the semantic role of each input token given a predicate.

Context Encoder The context encoder encodes input tokens with embeddings, which are then consumed by a bi-directional LSTM with 3 layers. The resulting hidden states are $\mathbf{H} = [\mathbf{h}_1, \ldots, \mathbf{h}_N]$, where $\mathbf{h}_t = [\overleftarrow{\mathbf{h}}_t; \overrightarrow{\mathbf{h}}_t]$, the concatenation of forward and backward hidden states.

Biaffine role scorer First, the predicate and candidate argument encodings are transformed through separate linear layers:

$$\mathbf{g}_p^{\text{pred}} = \text{ReLU}(\mathbf{W}^{\text{pred}}\mathbf{h}_p + \mathbf{b}^{\text{pred}}),$$
$$\mathbf{g}_a^{\text{arg}} = \text{ReLU}(\mathbf{W}^{\text{arg}}\mathbf{h}_a + \mathbf{b}^{\text{arg}}),$$

where p is the word index of the predicate, and a is the word index of a candidate argument word. Finally, the biaffine layer computes the score for each semantic role an argument candidate is able to take for predicate p:

$$\mathbf{s}_{pa} = \mathbf{g}_a^{\text{arg}\top}\mathbf{W}^{\text{role}}\mathbf{g}_p^{\text{pred}} + \mathbf{U}^{\text{role}}[\mathbf{g}_a^{\text{arg}}; \mathbf{g}_p^{\text{pred}}] + \mathbf{b}^{\text{role}},$$

where $\mathbf{W}^{\text{pred}}, \mathbf{b}^{\text{pred}}, \mathbf{W}^{\text{arg}}, \mathbf{b}^{\text{arg}}, \mathbf{W}^{\text{role}}, \mathbf{U}^{\text{role}}, \mathbf{b}^{\text{role}}$ are model parameters.

Hyperparameters The hidden state sizes for both the encoder and the scorer are set to 768. Only randomly initialized embeddings are used for this model. The batch size is set to 8 dialogues, which may include different numbers of predicates. The models are optimized with Adam with learning rate set to 2×10^{-5}. Dropout layers with dropout rate 0.1 are applied to all layer transitions. Model performance is evaluated on the development set twice every epoch, and training is stopped whenever there is no observed improvement in development F1 score in 10 evaluations starting at the 10-th epoch.

F.3 SST-2

Two kinds of models are used on these two datasets to evaluate the noise-robust losses: the simple BiLSTM models and the large ALBERT (Lan et al., 2019) models. The classification layers for both models are the same, but they have different encoders. The BiLSTM models use the same encoder as the model for eCSRL, which is a 3-layered BiLSTM, whereas the ALBERT models use the pretrained ALBERT base (v2) model as the encoder. The classification layer for both models is a simple one layer feedforward neural network.

Hyperparameters The hidden state sizes for all models are set to 768. The BiLSTM uses randomly initialized embeddings. The batch size is set to 128 for SST-2 with the BiLSTM encoder and 32 with the ALBERT encoder. The models are optimized with Adam with learning rate set to 2×10^{-5} for BiLSTM and 1×10^{-6} for ALBERT. Dropout layers with dropout rate 0.1 are applied to all layer transitions. Model performance is evaluated on the development set twice every epoch for the BiLSTM, and eight times for the ALBERT. Training is stopped whenever there is no observed improvement in development accuracy score in 20 evaluations starting at the 10-th epoch.

Warm-up epochs	Noise rates r		
	0.2	0.3	0.4
0 epochs	85.8(0.7)	78.6(1.5)	66.9(6.3)
5 epochs	86.2(0.1)	79.8(0.8)	67.1(3.3)
10 epochs	85.9(0.6)	78.9(0.8)	66.0(6.5)
15 epochs	85.6(1.0)	78.8(0.8)	66.2(1.7)

Table 5: Accuracy results from experiments with different warm-up cutoffs for the joint training framework. The loss used in these experiments is ER-GCE, the dataset is SST-2 and the corruption method is model-based.

G Development experiments with different warm-up cutoffs

Table 5 shows the development results for the joint training framework with different number of epochs for warm-up. This shows that when training with the joint training framework where the quality predictor and the main classifier are trained together, training the main classifier first for 5 epochs achieves the best performance.

H Model-based label corruption

We utilize a pretrained ALBERT (Lan et al., 2019) base model for model-based label corruption in order to simulate the automatic label generation process. A five-fold corruption process is used. We first split the concatenation of the training and the development datasets into five equal proportions, and further divide each proportion into a training, development and test set following an 3.9:0.1:1 split. For each proportion, a pretrained ALBERT base classifier is finetuned on the training set for two epochs and the model with the highest performance on the development set is saved as the labeler. Finally, an equal amount of instances in test from each label with the lowest confidence score from the label is chosen for corruption to reach a certain noise rate, with the gold labels swapped for a different label. This creates a different scenario in terms of how noise interacts with training, which can be seen in Table 3. At low noise rates, models trained with model-based noise are generally more accurate than models trained with uniform noise, indicating that the corrupted instances are generally located at the decision boundary with only limited negative influence on the majority of the test instances. As the noise rate increases, models trained with model-based noise see a quicker decline of performance than models trained with uniform noise, indicating that more harmful correlations between input and labels are created by model-based corruption process than the uniform corruption process.