Noisy Label Regularisation for Textual Regression

Yuxia Wang♠ **Timothy Baldwin**♠♡ **Karin Verspoor**♠♢

♠ The University of Melbourne, Melbourne, Victoria, Australia
♡ MBZUAI, Abu Dhabi, UAE
♢RMIT University, Melbourne, Victoria, Australia
yuxiaw@student.unimelb.edu.au
tb@ldwin.net karin.verspoor@rmit.edu.au

Abstract

Training with noisy labelled data is known to be detrimental to model performance, especially for high-capacity neural network models in low-resource domains. Our experiments suggest that standard regularisation strategies, such as weight decay and dropout, are ineffective in the face of noisy labels. We propose a simple noisy label detection method that prevents error propagation from the input layer. The approach is based on the observation that the projection of noisy labels is learned through memorisation at advanced stages of learning, and that the Pearson correlation is sensitive to outliers. Extensive experiments over real-world human-disagreement annotations as well as randomly-corrupted and data-augmented labels, across various tasks and domains, demonstrate that our method is effective, regularising noisy labels and improving generalisation performance.

1 Introduction

Modern deep neural networks (DNNs) have millions or billions of trainable parameters, far more than the number of examples they are trained on. To avoid over-fitting, they are heavily reliant on large-scale training, including data derived through methods such as self supervision, data augmentation, and self labelling (Devlin et al., 2019; Wei and Zou, 2019; Wang et al., 2020c). However, such methods inevitably introduce noise, through biased data, unnatural inputs, or incorrect labels.

Weak perturbations applied to inputs can improve model performance by forcing DNNs to learn noise-invariant latent representations (Tang and Eliasmith, 2010; Goodfellow et al., 2016). But training with noisy labels has been shown to be detrimental to generalisation performance across tasks including image classification (Tanaka et al., 2018), dialogue generation (Akama et al., 2020), and entity–relation extraction (Chen et al., 2020). DNNs fit noisy labels by "memorising" each example — over-fitting corrupted training sets, and yielding poor generalisation (Arpit et al., 2017).

Given this background, our focus in this paper is on how to alleviate memorisation and improve generalisation when training with noisy labels. Previous related work has proposed three directions: (1) regularisation techniques (Arpit et al., 2017); (2) augmenting the loss function with an explicit representation of the distribution of noise (Sukhbaatar et al., 2015; Patrini et al., 2017); and (3) explicit detection of noisy labels (Tanaka et al., 2018; Nguyen et al., 2020; Lee and Chung, 2020; Desmond et al., 2020). However, the vast majority of this work has focused on classification tasks, and there has been very little work in the context of regression tasks and low-resource domains. In this work, we fill the gap by targeting noisy label regularisation for text regression, in the form of semantic text similarity (STS), sentiment analysis, and machine translation quality assessment.

Our work makes three contributions: (1) empirical clarification of the role of explicit regularisation in noisy label training in a regression setting; (2) proposal of an effective noisy label detection method for continuous labels; and (3) extensive experiments across various regression tasks under both real-world and synthetic noisy labels, including state-of-the-art results on MedSTS. The code associated with this paper is available at: https://github.com/yuxiaw/Regularise-Regression-Noisy-Labels.

2 Related Work

2.1 Regularisation of Noisy Labels in Classification Settings

Regularisation: Explicit regularisation techniques such as dropout, weight decay, or data augmentation can help to alleviate over-fitting, improving model generalisation (Arpit et al., 2017; Tanaka et al., 2018). They do not, however, prevent classi-

Proceedings of the 29th International Conference on Computational Linguistics, pages 4228–4240
October 12–17, 2022.

fier degradation caused by noisy labels (Harutyunyan et al., 2020). Gradient descent with early stopping and its variants are provably robust to noisy labels (Li et al., 2020; Hu et al., 2020). While empirically verified to be effective for image classification, their performance in textual regression tasks is unknown.

Noise Distribution Matrix: An alternative approach is to correct the loss function with a noise distribution transition matrix (Sukhbaatar et al., 2015; Patrini et al., 2017; Yao et al., 2019; Tanno et al., 2019). Formally, let l and l^{GT} be the noisy and ground-truth labels. The noise transition matrix T is defined as $t_{ij} = p(l = j|l^{GT} = i)$, where the element of the i-th row and the j-th column t_{ij} represents the probability of mis-annotating golden class i to incorrect label j. The cross entropy loss is modified to $\mathcal{L}(\boldsymbol{\theta}, X, Y) = \frac{1}{n}\sum_{n=1}^{n} \log(\mathbf{y}_i^\intercal \mathbf{s}(\boldsymbol{\theta}, \mathbf{x}_i))$, where $\mathbf{s}(\boldsymbol{\theta}, \cdot)$ is the classifier. In classification tasks, the probability of misclassification between classes $p(l = j|l^{GT} = i)$ is well-defined. However, it is not clear how to define the matrix T for continuous output variables in a regression setting.

Noisy Label Detection has been explored extensively in classification settings, especially for images, but has received very little attention for textual regression problems. Noisy instances are typically identified based on model prediction (Zheng et al., 2020; Ye et al., 2021), such as comparing predicted labels (pseudo-labels) l^P with annotated labels l^A during training (Tanaka et al., 2018; Berthelot et al., 2019; Nguyen et al., 2020; Lee and Chung, 2020; Desmond et al., 2020), and the label distribution confidence (Liu et al., 2020). However, exact label match ($l^A = l^P$) is too strict a requirement for regression tasks. We relax the criterion to a range controllable by a threshold τ, i.e. $|l^A - l^P| < \tau$. This makes it identical to the loss-based criterion $(l^A - l^P)^2 < \tau$ in regression using mean-squared error loss (MSE). Specifically, instances that result in small loss can be considered to be clean (Shen and Sanghavi, 2019).

2.2 Detect Noisy Labels in Regression

To our knowledge, the only research addressing noisy labels in a textual regression setting is: (1) Wang et al. (2022), who select high-disagreement labels using the predictive variance of uncertainty models; and (2) Takamoto et al. (2020), who identify outliers based on the absolute difference between teacher model predictions and target labels.

Noise filtering also relates to data sampling in active learning. It aims to select the most informative/useful data points from an unlabelled pool, leveraging the least labelling effort to reach the best performance.

Sampling in Active learning: Regression tasks are also under-researched in the active learning literature (Elreedy et al., 2019; Zhang et al., 2020). Cai et al. (2013) sample data associated with the maximum gradient of the loss function, typically based on squared error, and Sugiyama (2006) aims to minimise the conditional expectation of the generalisation error. These are akin to the loss criteria in noisy label identification.

Separately, Wu (2019) considers representativeness and diversity in initial data collection and sequential query selection, and Wu and Huang (2022) select the most beneficial samples to label based on three emotion primitives: valence, arousal, and dominance for affect estimation. However, these methods are too domain-specific to adapt to general-purpose regression tasks.

3 Task and Datasets

In this paper, we investigate text regression across three separate tasks, and a total of 10 datasets.

3.1 Tasks

The three tasks we target in this research are STS, sentiment analysis, and machine translation quality estimation, which we outline below.

Semantic textual similarity (**STS**) assesses the degree of semantic equivalence between two (short) texts (Corley and Mihalcea, 2005). The aim is to predict a similarity score for a sentence pair $(S1, S2)$, generally in the range $[0, 5]$, where 0 indicates complete dissimilarity and 5 indicates equivalence in meaning. As an example:

S1: *Total minutes spent in timed codes: 10 mins.*
S2: *Total minutes spent in timed codes: 33 mins.*

is labelled 4, as the two texts differ only in very specific content (underlined).

Sentiment analysis (**SA**) rating involves predicting a sentiment score for a review S, in the range 1 (extremely negative) to 5 (extremely positive).

Machine translation quality estimation, based on the direct assessment (**DA**) approach (Graham et al., 2017), aims to predict a normalised quality score for text pair $(S1, S2)$, where $S2$ is machine translated from $S1$. As such, it is similar to STS, but differs in that it is cross-lingual.

3.2 Datasets

We evaluate on different-sized datasets across various domains for STS and SA, and two identically-sized datasets for DA, as summarised in Table 1.

For STS, we use: three large-scale general datasets — STS-B (Cer et al., 2017), SICK-R (Marelli et al., 2014), and STS-G (Wang et al., 2020c); and two small clinical data sets — MedSTS (Wang et al., 2018) and N2C2-STS (Wang et al., 2020a).

For SA, we use: a large-scale product review dataset — Yelp (Sabnis, 2018); and two small datasets of movie and paper reviews — 10Movie (Benlahbib, 2019) and PeerRead (Kang et al., 2018). We augment 10Movie with 700 examples from IMDB movie reviews (Maas et al., 2011) (see Appendix A.1 for details of the label conversion process), and also augment PeerRead with 399 Spanish paper reviews (Keith et al., 2017) which we automatically translate into English.

For DA, we employ two language pairs from WMT2020 (Specia et al., 2020), namely ru-en and ro-en, which are low- and medium-resource language pairs, respectively.

As evaluation metrics, we use Pearson's correlation (r) and Spearman's correlation (ρ) between the predicted and gold standard scores.

3.3 Notation and Loss Function

Throughout this paper, raw examples, column vectors, and matrices are denoted in lower-case italics, bold, and upper-case italics, respectively (e.g. x, $\mathbf{x}$ and X). $\boldsymbol{\theta}_{encoder}$ and $\boldsymbol{\theta}_{reg}$ represent parameters of the transformer encoder and task-specific regression layers, and $f(\boldsymbol{\theta}, \cdot)$ refers to the whole model. Assuming a dataset with N instances $\mathcal{D} = \{(x_1, y_1), \cdots, (x_i, y_i), \cdots, (x_N, y_N)\}$, where (x_i, y_i) is the ith instance of $\mathcal{D}$, $y_i \in [0, 5]$, $\mathbf{x}_i = s(\boldsymbol{\theta}_{encoder}, x_i)$ is the embedding of x_i. The loss function is the empirical risk of the mean square error (MSE): $\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N} (f(\boldsymbol{\theta}, x_i) - y_i)^2$.

4 Case Study

We first examine the susceptibility of DNNs to overfit random labels (Zhang et al., 2017), based on the clinical N2C2-STS data set using BERT (Devlin et al., 2019). Then we conduct ablation experiments using various regularisation techniques, to observe whether they can reduce the degradation caused by noisy labels.

Dataset	Size (Train, Test, Dev)	Range	Domain
SICK-R (2014)	4500, 4927, 500	$[1, 5]$	general
STS-B (2017)	5749, 1379, 1500	$[0, 5]$	general
STS-G (2020)	28518, —, —	$[0, 5]$	general
MedSTS (2018)	750, 318, —	$[0, 5]$	clinical
N2C2-STS (2019)	1642, 412, —	$[0, 5]$	clinical
Yelp (2018)	5000, —, —	[1,5]	product
10Movie+IMDB (2019)	1400, 300, —	[1,5]	movie
PeerRead+Spanish (2018)	1638, 290, —	[1,5]	paper
WMT ru-en (2020)	7000, 1000, 1000	$[0, 100]$	low-resource
WMT ro-en (2020)	7000, 1000, 1000	$[0, 100]$	med-resource

Table 1: STS/SA rating/DA datasets. "Train", "Test", "Dev" = number of text pairs; "Range" = label range. In practice, DA is normalised by z-scoring.

Hypothesis Arpit et al. (2017) empirically showed that explicit regularisation, especially dropout coupled with adversarial training, can reduce memorisation of noise without reducing a model's ability to learn. Zhang et al. (2017), on the other hand, argued that it is neither necessary nor sufficient for controlling generalisation error in deep learning. Overall, explicit regularisation may improve generalisation performance, but does not explicitly deal with noisy labels.

4.1 Experiment

Regression Model Structure: The regression model used here and in Section 6 takes the hidden state of the `[CLS]` token output for the single sentence or sentence pair from BERT, $\mathbf{h} \in R^d$. This is fed through a two-layer MLP, structured as:

$$\mathbf{h}' = \tanh(\mathbf{W}\mathbf{h} + \mathbf{b}) \quad (1)$$

$$\hat{y} = \mathbf{w}^\intercal \mathbf{h}' + b \quad (2)$$

where $\hat{y}$ is the predicted score, and $\mathbf{W} \in \mathbb{R}^{d \times d}$, $\mathbf{b}, \mathbf{w} \in \mathbb{R}^d$, and $b \in \mathbb{R}$ are trainable parameters of task-specific layers, denoted as "CLS-BERT".

Corrupted Training Set: To generate partially-noisy training data, we corrupt training set $\mathcal{D}$ by randomly selecting M instances and replacing their labels with $s \in [0, 5]$ sampled from a uniform distribution, forming noisy subset $\mathcal{D}_{noisy}$, leaving the clean partition $\mathcal{D}_{clean}$. Thus the corrupted training set is $\mathcal{D}' = \mathcal{D}_{noisy} \cup \mathcal{D}_{clean}$ (where $|\mathcal{D}| = |\mathcal{D}'| = N$).

Experimental Setup: We randomly split the 1,642 instances in the N2C2-STS training set into 1,242 and 400 instances, as training set $\mathcal{D}$ and a validation set. $M = \alpha \cdot N$ is decided by noise ratio $\alpha \in \{0.2, 0.4, 1.0\}$ to generate three corrupted training sets, denoted *corrupt2*, *corrupt4*, and *corrupt10*, respectively, with corresponding

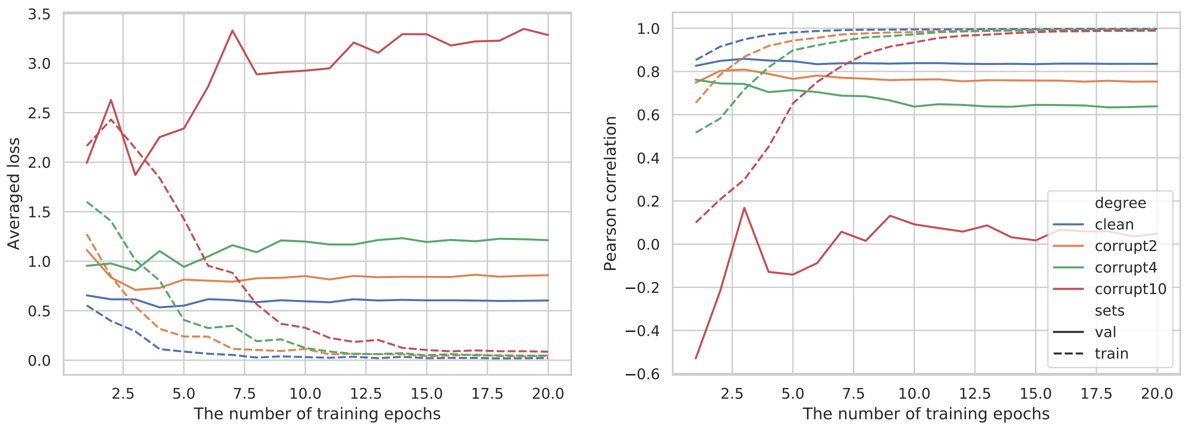

Figure 1: The loss (left) and r (right) on N2C2 train and validation sets over four different degrees of corrupted training set: clean ($\mathcal{D}$), corrupt2, corrupt4 and corrupt10.

clean partitions denoted as *clean2*, *clean4* and $\varnothing$ below. Note that the same validation set is used for all experiments in this section and Section 6.1.

The model is trained on *bert-base-uncased*, optimising with a linear scheduler with warmup proportion = 0.1, train batch size = 16, learning rate = 2e-5, and training epochs = 20.

4.1.1 Results

BERT is structured with layer normalisation and residual connections (He et al., 2016), noting that pre-training has been shown to be beneficial for generalisation by alleviating exposure bias. We are thus interested in whether using pre-trained BERT is robust to the effects of random noisy labels.

Using the baseline regression model, we fine-tune over the varyingly-corrupted training sets, each for 20 epochs. As per Figure 1, training with noisy labelled data is detrimental to generalisation performance, and more training exacerbates the effect, especially for noisier data.

As we increase the amount of noise, the training loss decreases and the model takes longer to converge. Particularly on fully-corrupted training set *corrupt10*, the training loss rises first and then starts to fit random labels, taking eight epochs to reach the same training accuracy as the *clean* set $\mathcal{D}$ in the first epoch. This shows that pre-trained BERT can reduce fitting to random labels early in training, and in general slows down convergence. However, since the random labels are fixed across epochs, iterating over the training set multiple times leads to (over)fitting the random labels perfectly.

4.2 Explicit Regularisation

We use the following regularisation techniques:

- **Early stop (“ES”)**: return the trained model where the lowest validation error is obtained (based on Pearson’s correlation).
- **Weight decay (“WD”)**: update parameters by $\theta_t = (1 - \beta)\theta_{t-1} - \alpha g_t$, where $\beta \in \{0.01, 0.05, 0.1\}$ is a weight decay coefficient, α is the learning rate, and g_t is the gradient at update step t.
- **Dropout (“DP”)**: replace Eq (1) with $\mathbf{h}' = \mathrm{dropout}(\tanh(\mathbf{W}\mathbf{h} + \mathbf{b}))$.
- **Data augmentation**: perform data augmentation via back translation (“BT”) or segment reordering (“SR” = randomly permute the order of segments separated by commas or semicolons) following Wang et al. (2020b).
- **Cross Domain Pre-fine-tuning**: fine-tune the model with general-purpose STS-B (“STSB”) training set for 3 epochs before fine-tuning on the clinical STS data.

4.2.1 Results

We present the results in Table 2. Comparing rows 1 and 2 (no regularisation vs. early stopping), early stopping improves performance over both clean and corrupted training data, especially on *corrupt4*. Therefore we combine it with the other strategies. Weight decay (WD) (rows 3–5) has negligible impact, but markedly improves *corrupt2* through dropout and data augmentation (rows 6–8 and 11–13), and *corrupt4* through back-translation (rows 7 and 12). Pre-fine-tuning provides large gains in accuracy on both clean and corrupted data sets (row 9), especially coupled with early stopping, weight decay and dropout (rows 10 and 14).

In sum, explicit regularisation improves generalisation performance, not just on corrupted data

ID	Train set setting	$\mathcal{D}$			corrupt2			corrupt4			clean2			clean4		
		r	ρ	loss	r	ρ	loss	r	ρ	loss	r	ρ	loss	r	ρ	loss
1	baseline	.835	.830	0.603	.754	.740	0.858	.639	.636	1.212	.845	.835	0.570	.836	.818	0.598
2	ES	.859	.837	0.614	.809	.797	0.710	.762	.724	0.953	.858	.835	0.529	.851	.822	0.592
3	ES + WD (0.01)	.857	.835	0.518	.810	.799	0.677	.762	.724	0.953	.854	.835	0.535	.852	.816	**0.532**
4	ES + WD (0.05)	.857	.835	0.613	.818	.805	0.670	.760	.722	0.955	.852	.831	0.559	.853	.818	0.544
5	ES + WD (0.1)	.857	.837	0.603	.808	.803	0.680	.769	.731	0.952	.852	.830	0.561	**.854**	**.825**	0.569
6	ES + DP	.858	.839	0.570	.831	.799	0.717	.758	.726	**0.849**	.857	.828	0.519	.840	.818	0.626
7	ES + BT	.858	.838	0.526	.833	.810	0.673	.775	.734	1.124	.850	.825	0.551	.847	.823	0.650
8	ES + SR	.855	.836	0.526	.832	.796	0.615	.761	.749	0.999	**.858**	**.838**	0.541	.849	.823	0.630
9	ES + STSB	.867	.842	0.491	.846	.817	0.745	.783	.778	0.890	.857	.823	0.533	.852	.822	0.571
10	ES + WD (0.1) + STSB	**.867**	**.843**	**0.488**	**.850**	.822	0.724	.783	.778	0.890	.857	.823	0.531	.852	.822	0.571
11	ES + DP + WD (0.1)	.853	.835	0.588	.837	.801	0.679	.756	.737	1.167	.855	.823	**0.517**	.842	.820	0.622
12	ES + DP + WD (0.1) + BT	.859	.833	0.508	.833	.802	0.631	**.796**	.763	0.949	.843	.830	0.587	.839	.817	0.643
13	ES + DP + WD (0.1) + SR	.853	.826	0.538	.833	.787	**0.589**	.790	.763	1.080	.856	.835	0.524	.834	.812	0.627
14	ES + DP + WD (0.1) + STSB	.864	.841	0.520	.848	**.823**	0.731	.779	**.780**	0.943	.853	.822	0.561	.845	.813	0.589

Table 2: Averaged loss, r and ρ on N2C2 validation set with various combinations of regularisation techniques ("ES" = early stopping, "WD" = weight decay, "DP" = dropout, "BT" = data augmentation with back translation, "SR" = data augmentation with segment reordering, "STSB" = fine-tuning over STS-B), over five different training sets ("$\mathcal{D}$" = all clean, "corrupt2" = 0.2 corrupted, "corrupt4" = 0.4 corrupted, "clean2" = clean complementary set of corrupt2, "clean4" = clean complementary set of corrupt4). The best result in each column is bolded.

but on clean data as well. The fact that the best accuracy is still much worse than training on fully clean $\mathcal{D}$, and also worse than on their clean components, clean2 and clean4, also confirms that explicit regularisation can't control the generalisation error caused by noisy labels. It additionally suggests that removing noisy examples could lead to improvements, which we verify in Section 5.

5 Noisy Label Detection

We propose a two-step method to identify noisy labels from training data based on iterative predictions, followed by three different strategies for training with noisy examples, namely: (1) **DIS**: discard noisy examples; (2) **REP**: repair noisy labels with pseudo labels; and (3) **RES**: resample the same number of instances from the "clean" set, to make up for discarded noisy examples.

5.1 Prediction criterion

We relax the requirement of exact match between the pseudo label $\hat{p}$ and the annotated label y by measuring the absolute difference, and them to match if the difference is within a predefined range $|y - \hat{p}| \leq \tau$, which is a tuneable hyper-parameter. The pseudo label is obtained by averaging predictions over multiple training iterations (see line 9 of Algorithm 1).

If τ is small, precision will be low and recall of noisy instances will be high, whereas if τ is large, recall will be low, but precision will be high, negatively affecting training quality. To achieve a balance between precision and recall, we use Pearson's correlation to assess the "noisiness" of each noisy candidate, where a relatively small τ ensures high recall.

Algorithm 1 Train on Noisy Labelled Data

```
1:  Input: Training and validation set $\mathcal{D}_{train}$, $\mathcal{D}_{val}$
2:  $\mathcal{D}_{clean} \leftarrow \mathcal{D}_{train}$
3:  for $i$ in range(1, $epochs$) do
4:      $M_i \leftarrow train(\mathcal{D}_{clean})$
5:      if $acc(M_i, \mathcal{D}_{val}) \geq acc(M_{best}, \mathcal{D}_{val})$ then
6:          $M_{best} \leftarrow M_i$
7:      end if
8:      $\hat{\mathbf{y}}_i \leftarrow M_{best}(\mathbf{x})$
9:      obtain pseudo labels $\hat{\mathbf{p}}_i \leftarrow \frac{1}{i}\sum_{j=0}^{i} \hat{\mathbf{y}}_j$
10:     get noisy candidate set by Prediction Criterion $f_1$
11:     $\mathcal{C}_{clean}, \mathcal{C}_{noise} = f_1(\mathcal{D}_{train}, \hat{\mathbf{p}}_i, \tau)$
12:     determine noisy set by Pearson Criterion $f_2$
13:     $\mathcal{D}_{clean}, \mathcal{D}_{noise} = f_2(\mathcal{C}_{clean}, \mathcal{C}_{noise}, m, K, \varepsilon)$
14:     $\mathcal{D}_{clean}$ by Repairing
15:     $\mathcal{D}_{clean} \leftarrow$ repair$(\mathcal{D}_{noise}, \hat{\mathbf{p}}_i) \cup \mathcal{D}_{clean}$
16:     $\mathcal{D}_{clean}$ by Resampling
17:     $\mathcal{D}_{clean} \leftarrow$ sample$(\mathcal{D}_{clean}, size(\mathcal{D}_{noise})) \cup \mathcal{D}_{clean}$
18: end for
19: Output: $M_{best}$
```

5.2 Pearson correlation criterion

In Table 2, we found that the smallest loss did not always mean the best performance r. The former is measured for an individual example, while the latter considers correlation of the combined set of instances. We therefore propose to use a summary evaluation metric for selection of noisy instances, adopting Pearson's correlation (r) due to its sensitivity to outliers (Wilcox, 2004; Chok, 2010; Mathur et al., 2020). That is, Pearson correlation is strongly sensitive to linear relationships: r is maximised when two variables are linearly related to each other, whereas Spearman correlation is maximised when two variables are monotonically related, whether the relationship is linear or not. A single outlier influences the results of r (Rousselet and Pernet, 2012).

Under this criterion, for each noisy candidate (x, y) identified by the prediction criterion (line 13), we randomly select m clean examples from the clean subset $\mathcal{C}_{clean}$ (the training set with noisy candidates filtered out) and calculate the correlation r, then add (x, y) to this set and calculate the new correlation as r'. If a large perturbation is observed — i.e., $|r - r'| > \varepsilon$, e.g. > 0.01 — x is considered to be noisy; otherwise it is considered to be clean.

However, when most of the sampled labels are clustered together with no obvious relationship, r tends to be unjustified, and when points are distributed uniformly, the score is fair (see Figure 1 in Rousselet and Pernet (2012)). This is attributed to the ordinary least square solution: one badly positioned point can have a dramatic influence on the results (Hubert et al., 2008). This instability leads to a less powerful statistical test. To smooth the number (denote as A), we project A of the lower-order into higher-order A^K by repeating this process K times. This can mitigate the large variance and inconsistency in the correlation (Song et al., 2021). We make the final decision by voting: only if all K votes agree that the given training instance is noisy is it removed.

Though the first training epoch is trained on the whole corrupted training set, it does not impact generalisation significantly, because memorised features are not learned in the early stages of training. It also benefits domains with limited training data. That is, even when initialising with pre-trained weights, the STS model is not accurate enough to filter noisy labels accurately in the first iteration, leading to a high percentage of instances being filtered out and exacerbating data sparsity.

5.3 Time Complexity

In terms of computational efficiency on large-scale datasets, despite the two-step detection process and repeatedly calculating Pearson's correlation criterion K times, time complexity varies linearly — $\mathcal{O}(N) \times K$. This is negligible when compared with the training time.

6 Experiments

We first evaluate the noise detection method on two synthetically-corrupted versions of N2C2-STS ("N2C2") (*corrupt2* and *corrupt4*), where we have perfect knowledge of the noisy and clean subset, and then apply our methods to real-world STS, SA, and DA datasets, where noisy labels are unknown.

6.1 Train on Randomly Corrupted Data

Setup We employ the optimal combination of regularisation methods from Section 4 as a strong baseline, namely row 14 of Table 2 (early stopping + dropout + weight decay + pre-fine-tuning), and set $wd = 0.01$, $tolerance = 0.75$, $threshold_r = 0.01$, $K = 5$, and $m = 8$ for noise detection. Other hyper-parameters are as per Section 4. Experimental results are averaged over ten runs based on ten random seeds to account for variance for small test sets.[1]

Result We highlight three findings from Table 3: (1) on randomly corrupted labelled data, noise filtering improves validation performance, particularly for high-degree corruption, decreasing the generalisation error by a large margin; (2) the strategies of DIS and RES perform largely the same, better than REP, so we use DIS in most cases in our following experiments; and (3) precision and recall at noise detection impact on the end-task performance, while the second step of Pearson correlation-based filtering critically improves precision. For example, after the first training epoch on corrupt4 with "discard", precision is 53.82% and recall is 68.15%, and with the Pearson correlation filtering the precision improves to 65.98%, leading to the improvement in row 3.[2]

Discarding automatically-detected noisy labels can maintain improved performance after fast convergence. Considering *corrupt4* in Figure 2, we observe that training with noisy labels for more iterations reduces the validation performance, and the loss also peaks. Through discarding noisy examples by our method during training, the performance can be improved and maintained, and the loss correspondingly drops. This finding is especially vital in the absence of a clean validation set. In such cases, early stopping becomes invalid,

[1] For reproduction purposes, we use seeds = [30 32 40 42 43 45 90 101 3405 3407] for all small test sets.

[2] Precision and recall can be obtained only in the unrealistic scenario of the noisy subset being explicitly known, and are unavailable for real-world data.

	criterion	corrupt2			corrupt4		
		r	ρ	loss	r	ρ	loss
Base	NA	0.841 ± 0.005	0.815 ± 0.006	0.661 ± 0.048	0.797 ± 0.019	0.785 ± 0.009	0.911 ± 0.102
Dis	prediction	0.844 ± 0.004	0.820 ± 0.006	0.628 ± 0.068	0.808 ± 0.012	0.791 ± 0.011	0.818 ± 0.121
Dis	two-step	**0.847** ± 0.004	**0.824** ± 0.006	**0.586** ± 0.038	**0.822** ± 0.006	**0.803** ± 0.006	0.687 ± 0.095
Rep	two-step	0.842 ± 0.005	0.816 ± 0.006	0.655 ± 0.053	0.799 ± 0.018	0.782 ± 0.013	0.861 ± 0.092
Res	two-step	0.843 ± 0.006	0.822 ± 0.007	0.593 ± 0.044	**0.822** ± 0.016	0.798 ± 0.013	**0.653** ± 0.043

Table 3: Results on N2C2 validation set trained on partially corrupted N2C2 train sets: *corrupt2* and *corrupt4* under three noisy label training strategies; "prediction" means we only use the first-step criterion; Base = baseline.

	ro-en dev			ro-en test			ru-en dev			ru-en test		
	r	ρ	loss	r	ρ	loss	r	ρ	loss	r	ρ	loss
Base	0.834	0.791	**0.309**	0.832	0.778	**0.290**	0.629	0.612	**0.512**	0.645	0.612	**0.531**
Dis	**0.841**	**0.807**	0.319	**0.838**	**0.793**	0.292	0.640	**0.617**	0.576	0.653	0.627	0.599
Res	0.838	0.801	0.328	0.837	0.789	0.299	**0.644**	0.612	0.538	**0.660**	**0.630**	0.554

Table 4: Results for DA-style quality estimation over the two WMT language pairs.

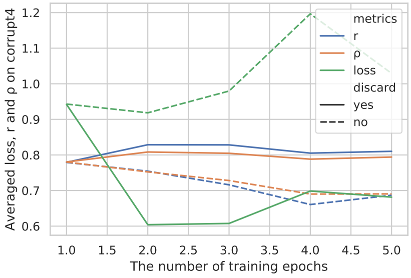

Figure 2: Averaged loss, r, and ρ on N2C2 validation set with (solid line) and without (dotted line) discarding noisy labels using *corrupt4* over the first five epochs.

and we generally train for a constant number of iterations. Identifying and filtering noisy labels can prevent the continuous decline.

6.2 Training on Real-world Noisy Labels

Real-world label noise is a natural outcome of the dataset collection process, and emanates from three primary sources (Algan and Ulusoy, 2021): (1) conflicting opinions of multiple annotators due to diverse interpretations and varying level of expertise, e.g. machine translation quality assessment; (2) inherent uncertainty due to domain complexity such as in the clinical domain; and (3) to collect large amounts of data, textual regression tasks tend to resort to various data augmentation strategies, which is known to result in noisy labels.

6.2.1 Human Disagreement Labels

DA datasets contain examples with highly ambiguous labels due to its subjectivity and the nature of language ambiguity (Wang et al., 2022). Disagreements among annotators very often persist even if more ratings are collected and more context is provided to the raters (Pavlick and Kwiatkowski, 2019). To evaluate the effectiveness of identifying high-disagreement labels, we perform our noise detection method on two DA language pairs: ru-en and ro-en.

Setup: We fine-tune BERT based on *bert-base-multilingual-cased* with maximum sequence length of 128 for five epochs. Hyper-parameter $tolerance$ is set as 1.0 in Dis for both, and 0.75 for ru-en in Res based on the validation set. Other settings are the same as Section 6.1.

Results: In Table 4, over both ru-en and ro-en, employing either discarding or resampling can improve the correlation r/ρ by more than one point on average, indicating that the approach can filter high-ambiguity labels that confuse the model, thus boosting accuracy. It also shows that the lowest loss does not always result in the best performance in terms of r/ρ.

6.2.2 Complicated Domain Labels

In clinical STS, some examples are complicated for clinicians to reach consensus, leading to low inter-annotator agreement, such as Cohen's $\kappa = 0.6$ for N2C2 training labels. In contrast to the disagreements mentioned above, the issue is more varying levels of domain knowledge and the inherent complexity of mapping textual similarity onto a single scalar. Additionally, the N2C2 ground-truth lacks an adjudication process, and gold scores are de-

ID	Train Data	PreF	DIS	r	ρ	loss
1	N2C2 train	NA	NA	0.860 ± 0.008	0.805 ± 0.009	0.759 ± 0.038
2	N2C2 train	stsg	No	0.860 ± 0.003	0.816 ± 0.005	0.746 ± 0.017
3	N2C2 train	stsg	Yes	0.878 ± 0.010	0.810 ± 0.007	**0.585** ± 0.077
4	SR train	stsg	No	0.868 ± 0.005	0.816 ± 0.009	0.825 ± 0.035
5	SR train	stsg	Yes	**0.885** ± 0.010	**0.824** ± 0.014	0.595 ± 0.082

Table 5: Averaged loss, r/ρ on the N2C2 test set **w/o** noise-filtering using N2C2 train and segment-reordered N2C2 train based on CLS-BERT pre-fine-tuned (PreF) on large-scale general STS corpus STS-G.

rived by simple averaging of scores from two annotators (Mahajan et al., 2020). We aim to recognise strong-disagreement labels from the original N2C2 training data and the segment reordered version (see Section 4.2), to improve generalisation.

Setup: We employ a larger-scale general-purpose STS labelled dataset STS-G (Wang et al., 2020c) to learn a general-domain STS model. Other settings are the same as Section 6.1, except for training epoch=3.

Results: Comparing rows 1 and 2 in Table 5, pre-fine-tuning on STS-G doesn't improve performance, and just diminishes the standard deviation. By applying noise-filtering, the loss decreases and correlation improves appreciably (row 3). We speculate that STS-G is large enough to capture general STS task properties, so multiple iterations of training on the N2C2-STS training set don't lead to any gains, but stabilise at a local minimum. Noisy label training helps escape the local minimum by filtering suspected examples, bringing about the large drop in loss, and boosting performance. Fine-tuning over segment-reordered training data augments corruption, making noisy labels more noticeable and easier to detect, thus resulting in the best r/ρ (row 5).

Analysis: Can our method recognise high-disagreement labels? Is detection more accurate with segment-reordered text? As noisy labels are unknown, and individual labels from different annotators are not available either, we manually analysed the first 400 instances in the training set, finding 44 labels that we don't agree with, 16 of which overlap with the detected noisy labels with N2C2 original training data after the first epoch (prec.=0.43, recall=0.36). In the setting of SR, we observed the overlap with our annotations increases to 24 (prec.=0.43, recall=0.55), so the final result improves with an increase in detection accuracy.

6.2.3 Data-Augmented Labels

We apply our method to denoising instances generated through data augmentation, either through synthetic generation based on annotation guidelines, or through rule-based conversion.

Clinical STS with Synthetic Labels Wang et al. (2020b) show that a hierarchical convolutional network based on BERT ("HConvBERT"), in which the BERT-base bottom eight encoder layers are frozen and only parameters of the top four layers are updated, is beneficial to training over a small-scale data set. Since the size of clinical STS datasets is small, we experiment with HConvBERT (hconv) in addition to CLS-BERT (cls).

Synthetic Data Generation: To generate clinical sentences, following Wang et al. (2020c), we sample discharge summaries from MIMIC-III (Johnson et al., 2016) and segment them into 27 parts based on section subtitles (topics). We select the topics of medications, illnesses, diagnoses, and follow-up instructions at which the clinical proxy is not expert, and split into sentences. As medication-related examples emphasise specific medication names, different rules are applied than for other topics. All medication sentences are grouped by medication name; this is always their first word. If S1 and S2 are sampled from the same name group, a similarity score of 3 is assigned, with +0.5 for every increase of 0.2 in $l/L1$ and $l/L2$, where l is the number of shared tokens between S1 in length of L1 and S2 in L2. Otherwise, it is labelled as 1.

For other topics, when sampling sentences from two different topics, the label is set to the range $[0, 1]$. Sentences sampled from the same topic are theoretically in the range $[1, 5]$, but in practice are generally in the range $[1, 2]$ because high similarity under random pairing is unlikely. To obtain pairs in the range $[2, 4.5]$, we randomly sample two sentences from the same topic, and use one as a "template" (S1). We then randomly replace a sequence of d words in the template with the same position (word index) sequence of the second sentence, forming S2. This pair (S1,S2) is labelled as 4.5 if only 10% words are replaced in S1, i.e. $d/L1$=0.1, and 4, 3.5, 3 and 2 with more words replaced, $d/L1$=0.2, 0.3, 0.4 and 0.5, respectively.

In total, we generate 1534 cases ("syn1534"), including 416 medication cases (200 in range $[1, 2]$ and 416 in range $[3, 5]$) and 1118 cases of other topics (200 in range $[0, 1]$, 200 in range $[1, 2]$ and 718 in range $[2, 5]$).

Train Data	Model	PreF	DIS	r	ρ	loss
N2C2 train	cls	NA	NA	0.860 ± 0.008	0.805 ± 0.009	0.759 ± 0.038
N2C2 train	cls	stsb	NA	0.852 ± 0.006	0.813 ± 0.007	0.792 ± 0.054
+ syn1534	cls	stsb	No	0.854 ± 0.004	0.799 ± 0.004	0.788 ± 0.039
+ syn1534	cls	stsb	Yes	0.868 ± 0.003	0.785 ± 0.005	0.669 ± 0.040
N2C2 train	hconv	stsb	NA	0.872 ± 0.003	**0.828** ± 0.003	0.717 ± 0.026
+ syn1534	hconv	stsb	No	0.867 ± 0.002	0.817 ± 0.005	0.760 ± 0.018
+ syn1534	hconv	stsb	Yes	**0.882** ± 0.003	0.805 ± 0.007	**0.669** ± 0.076
MedSTS train	cls	NA	NA	0.833 ± 0.004	0.764 ± 0.006	0.411 ± 0.024
MedSTS train	hconv	stsb	NA	0.850 ± 0.001	0.784 ± 0.004	0.372 ± 0.012
+ syn700	hconv	stsb	No	0.856 ± 0.002	0.789 ± 0.002	0.360 ± 0.019
+ syn700	hconv	stsb	Yes	**0.858** ± 0.003	**0.801** ± 0.006	**0.357** ± 0.026

Table 6: Averaged loss, r and ρ on the N2C2 test set (upper half) and MedSTS test set (bottom half) **w/o** noise-filtering trained *w/o* synthetic data.

	r	ρ	loss
Paper			
BASE	0.664 ± 0.007	0.664 ± 0.006	1.203 ± 0.036
FT	0.674 ± 0.003	0.679 ± 0.003	1.223 ± 0.035
FT+DIS	**0.681** ± 0.006	**0.686** ± 0.005	**1.192** ± 0.025
Movie			
BASE	0.791 ± 0.010	0.760 ± 0.006	0.533 ± 0.024
FT	0.810 ± 0.006	0.774 ± 0.005	**0.510** ± 0.018
FT+DIS	**0.815** ± 0.005	**0.777** ± 0.006	0.526 ± 0.032

Table 7: r/ρ on PeerRead (top) and 10Movie (bottom) test set **w/o** noise-filtering using PeerRead+Spanish and 10Movie+IMDB for paper and movie domains based on HConvBERT fine-tuned on Yelp (BASE), FT=fine-tune.

The STS model is first fine-tuned on STS-B to capture general-domain effects, then on the N2C2 training set combined with syn1534, and the MedSTS training set combined with a random sub-set of syn1534 of size 700 (“syn700”), to match the size of the MedSTS training set (750).

Results: Table 6 shows that combining synthetically-generated sentence pairs with gold-standard training data can improve performance, and discarding noisy labels results in further gains in accuracy on both N2C2-STS (upper half) and MedSTS (bottom half). We exceed previous state-of-the-art results on MedSTS $r = 0.848$ (Peng et al., 2019) $\rightarrow r = 0.858$. Further, pre-trained HConvBERT performs better than CLS-BERT.

Domain SA rating with Converted Labels Additionally, we evaluate denoising on SA rating. Two small-scale SA datasets in the academic paper and movie domains are augmented through rule-based conversion and machine translation (see Section 3), which inevitably introduce noise into the training sets.

Following the experimental setting of clinical STS with corrupted labels, we first fine-tune the regressor using the large-scale Yelp dataset (5,000 instances) based on HConvBERT, referred to as “HConvYelp”, then adapt it to the respective datasets by continuous fine-tuning combined with noise filtering. The results in Table 7 show that employing noise-filtering consistently improves performance for both datasets, particularly for the paper reviews, which are most domain-removed from the product reviews.

6.3 Overall Take-away

Through extensive experiments over randomly-corrupted and real-world noisy labels, we have demonstrated that our denoising method is effective at preventing memorisation, regularising noisy labels and improving generalisation performance on various regression tasks. The limitation of our method is that, while it is effective at detecting extreme outliers, it struggles to detect instances with weak disagreement, due to the fact that Pearson’s correlation is stable over distributions with moderate skewness (Chok, 2010). As such, it shows more impressive improvement in knowledge-rich domains like clinical notes and academic papers than general-purpose domains (see Section A.2).

7 Conclusion

Regularisation strategies improve model generalisation performance in a range of contexts, but are not able to effectively address generalisation degradation caused by training with noisy labels. In this paper, we have proposed a noisy label training method for text regression tasks, based on identifying noise through iterative prediction and targeted evaluation criteria, followed by discarding or repairing of noisy labels. Extensive experiments on three rating tasks demonstrate the effectiveness of our approach.

Acknowledgements

We thank the three anonymous reviewers for their helpful comments, and Yimeng Chen and Minghan Wang for various discussions. Yuxia Wang is supported by scholarships from The University of Melbourne and China Scholarship Council (CSC).

References

Reina Akama, Sho Yokoi, Jun Suzuki, and Kentaro Inui. 2020. Filtering noisy dialogue corpora by connectivity and content relatedness. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 941–958, Online. Association for Computational Linguistics.

Görkem Algan and Ilkay Ulusoy. 2021. Image classification with deep learning in the presence of noisy labels: A survey. *Knowledge Based Systems*, 215:106771.

Devansh Arpit, Stanislaw Jastrzebski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S. Kanwal, Tegan Maharaj, Asja Fischer, Aaron C. Courville, Yoshua Bengio, and Simon Lacoste-Julien. 2017. A closer look at memorization in deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, pages 233–242. PMLR.

Abdessamad Benlahbib. 2019. 1000 movie reviews (review + attached rating + sentiment polarity) for reputation generation. `https://data.mendeley.com/datasets/38j8b6s2mx/1`.

David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. 2019. Mixmatch: A holistic approach to semi-supervised learning. In *Advances in Neural Information Processing Systems*, pages 5049–5059.

Wenbin Cai, Ya Zhang, and Jun Zhou. 2013. Maximizing expected model change for active learning in regression. In *2013 IEEE 13th International Conference on Data Mining*, pages 51–60.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada.

Daoyuan Chen, Yaliang Li, Kai Lei, and Ying Shen. 2020. Relabel the noise: Joint extraction of entities and relations via cooperative multiagents. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5940–5950, Online. Association for Computational Linguistics.

Nian Shong Chok. 2010. *Pearson's versus Spearman's and Kendall's correlation coefficients for continuous data*. Ph.D. thesis, University of Pittsburgh.

Courtney Corley and Rada Mihalcea. 2005. Measuring the semantic similarity of texts. In *Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment*, pages 13–18. Association for Computational Linguistics.

Michael Desmond, Catherine Finegan-Dollak, Jeff Boston, and Matt Arnold. 2020. Label noise in context. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 157–186, 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.

Dina Elreedy, Amir F. Atiya, and Samir I. Shaheen. 2019. A novel active learning regression framework for balancing the exploration-exploitation trade-off. *Entropy*, 21(7):651.

Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. 2016. *Deep Learning*, volume 1. MIT press Cambridge.

Yvette Graham, Timothy Baldwin, Alistair Moffat, and Justin Zobel. 2017. Can machine translation systems be evaluated by the crowd alone. *Natural Language Engineering*, 23(1):3–30.

Hrayr Harutyunyan, Kyle Reing, Greg Ver Steeg, and Aram Galstyan. 2020. Improving generalization by controlling label-noise information in neural network weights. In *Proceedings of the 37th International Conference on Machine Learning*, pages 4071–4081. PMLR.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778.

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

Mia Hubert, Peter J Rousseeuw, and Stefan Van Aelst. 2008. High-breakdown robust multivariate methods. *Statistical Science*, 23(1).

Alistair EW Johnson, Tom J Pollard, Lu Shen, H Lehman Li-Wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. 2016. MIMIC-III: a freely accessible critical care database. *Scientific Data*, 3(1):1–9.

Dongyeop Kang, Waleed Ammar, Bhavana Dalvi, Madeleine van Zuylen, Sebastian Kohlmeier, Eduard Hovy, and Roy Schwartz. 2018. A dataset of peer reviews (PeerRead): Collection, insights and NLP applications. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1647–1661, New Orleans, Louisiana. Association for Computational Linguistics.

Brian Keith, Exequiel Fuentes, and Claudio Meneses. 2017. A hybrid approach for sentiment analysis applied to paper. In *Proceedings of ACM SIGKDD Conference*.

Jisoo Lee and Sae-Young Chung. 2020. Robust training with ensemble consensus. In *International Conference on Learning Representations*.

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

Yun-Peng Liu, Ning Xu, Yu Zhang, and Xin Geng. 2020. Label distribution for learning with noisy labels. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 2568–2574. ijcai.org.

Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150.

Diwakar Mahajan, Ananya Poddar, Jennifer J Liang, Yen-Ting Lin, John M Prager, Parthasarathy Suryanarayanan, Preethi Raghavan, and Ching-Huei Tsou. 2020. Identification of semantically similar sentences in clinical notes: Iterative intermediate training using multi-task learning. *JMIR Medical Informatics*, 8(11):e22508.

Marco Marelli, Luisa Bentivogli, Marco Baroni, Raffaella Bernardi, Stefano Menini, and Roberto Zamparelli. 2014. SemEval-2014 task 1: Evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 1–8, Dublin, Ireland. Association for Computational Linguistics.

Nitika Mathur, Timothy Baldwin, and Trevor Cohn. 2020. Tangled up in BLEU: Reevaluating the evaluation of automatic machine translation evaluation metrics. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4984–4997, Online. Association for Computational Linguistics.

Duc Tam Nguyen, Chaithanya Kumar Mummadi, Thi Phuong Nhung Ngo, Thi Hoai Phuong Nguyen, Laura Beggel, and Thomas Brox. 2020. Self: Learning to filter noisy labels with self-ensembling. In *International Conference on Learning Representations*.

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.

Ellie Pavlick and Tom Kwiatkowski. 2019. Inherent disagreements in human textual inferences. *Transactions of the Association for Computational Linguistics*, 7:677–694.

Yifan Peng, Shankai Yan, and Zhiyong Lu. 2019. Transfer learning in biomedical natural language processing: An evaluation of BERT and ELMo on ten benchmarking datasets. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 58–65, Florence, Italy. Association for Computational Linguistics.

Guillaume A Rousselet and Cyril R Pernet. 2012. Improving standards in brain-behavior correlation analyses. *Frontiers in Human Neuroscience*, 6:119.

Omkar Sabnis. 2018. Yelp review dataset. `https://www.kaggle.com/omkarsabnis/yelp-reviews-dataset`.

Yanyao Shen and Sujay Sanghavi. 2019. Learning with bad training data via iterative trimmed loss minimization. In *International Conference on Machine Learning*, pages 5739–5748. PMLR.

Yurun Song, Junchen Zhao, and Lucia Specia. 2021. Sentsim: Crosslingual semantic evaluation of machine translation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3143–3156.

Lucia Specia, Frédéric Blain, Marina Fomicheva, Erick Fonseca, Vishrav Chaudhary, Francisco Guzmán, and André F. T. Martins. 2020. Findings of the WMT 2020 shared task on quality estimation. In *Proceedings of the Fifth Conference on Machine Translation*, pages 743–764, Online. Association for Computational Linguistics.

Masashi Sugiyama. 2006. Active learning in approximately linear regression based on conditional expectation of generalization error. *Journal of Machine Learning Research*, 7:141–166.

Sainbayar Sukhbaatar, Joan Bruna, Manohar Paluri, Lubomir Bourdev, and Rob Fergus. 2015. Training convolutional networks with noisy labels. In *International Conference on Learning Representations*.

Makoto Takamoto, Yusuke Morishita, and Hitoshi Imaoka. 2020. An efficient method of training small models for regression problems with knowledge distillation. In *3rd IEEE Conference on Multimedia Information Processing and Retrieval*, pages 67–72. IEEE.

Daiki Tanaka, Daiki Ikami, Toshihiko Yamasaki, and Kiyoharu Aizawa. 2018. Joint optimization framework for learning with noisy labels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5552–5560.

Yichuan Tang and Chris Eliasmith. 2010. Deep networks for robust visual recognition. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 1055–1062.

Ryutaro Tanno, Ardavan Saeedi, Swami Sankaranarayanan, Daniel C. Alexander, and Nathan Silberman. 2019. Learning from noisy labels by regularized estimation of annotator confusion. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 11244–11253. Computer Vision Foundation / IEEE.

Yanshan Wang, Naveed Afzal, Sunyang Fu, Liwei Wang, Feichen Shen, Majid Rastegar-Mojarad, and Hongfang Liu. 2018. MedSTS: a resource for clinical semantic textual similarity. *Language Resources and Evaluation*, pages 1–16.

Yanshan Wang, Sunyang Fu, Feichen Shen, Sam Henry, Ozlem Uzuner, and Hongfang Liu. 2020a. The 2019 N2C2/OHNLP track on clinical semantic textual similarity: Overview. *JMIR Medical Informatics*, 8(11):e23375.

Yuxia Wang, Daniel Beck, Timothy Baldwin, and Karin Verspoor. 2022. Uncertainty estimation and reduction of pre-trained models for text regression. *Transactions of the Association for Computational Linguistics*, 10:1–17.

Yuxia Wang, Fei Liu, Karin Verspoor, and Timothy Baldwin. 2020b. Evaluating the utility of model configurations and data augmentation on clinical semantic textual similarity. In *Proceedings of the 19th SIGBioMed Workshop on Biomedical Language Processing*, pages 105–111, Online. Association for Computational Linguistics.

Yuxia Wang, Karin Verspoor, and Timothy Baldwin. 2020c. Learning from unlabelled data for clinical semantic textual similarity. In *Proceedings of the 3rd Clinical NLP Workshop*, Online. EMNLP.

Jason Wei and Kai Zou. 2019. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. 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 6381–6387, Hong Kong, China.

Rand Wilcox. 2004. Inferences based on a skipped correlation coefficient. *Journal of Applied Statistics*, 31(2):131–143.

Dongrui Wu. 2019. Pool-based sequential active learning for regression. *IEEE Transactions of Neural Networks Learning Systems*, 30(5):1348–1359.

Dongrui Wu and Jian Huang. 2022. Affect estimation in 3d space using multi-task active learning for regression. *IEEE Transactions of Affective Computing*, 13(1):16–27.

Jiangchao Yao, Hao Wu, Ya Zhang, Ivor W. Tsang, and Jun Sun. 2019. Safeguarded dynamic label regression for noisy supervision. In *The Thirty-Third AAAI Conference on Artificial Intelligence*, pages 9103–9110.

Shuquan Ye, Dongdong Chen, Songfang Han, and Jing Liao. 2021. Learning with noisy labels for robust point cloud segmentation. In *2021 IEEE/CVF International Conference on Computer Vision*, pages 6423–6432. IEEE.

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

Hongjing Zhang, S. S. Ravi, and Ian Davidson. 2020. A graph-based approach for active learning in regression. In *Proceedings of the 2020 SIAM International Conference on Data Mining*, pages 280–288. SIAM.

Songzhu Zheng, Pengxiang Wu, Aman Goswami, Mayank Goswami, Dimitris N. Metaxas, and Chao Chen. 2020. Error-bounded correction of noisy labels. In *Proceedings of the 37th International Conference on Machine Learning*, pages 11447–11457. PMLR.

A Appendix

A.1 Collecting Labels for Regression

Unlike computer vision tasks, which can make use of messy user tags or search engines and social media, it's hard to obtain usable continuous labels for semantic understanding tasks. Textual regression tasks tend to resort to data augmentation strategies or synthetic generation approaches to obtain labels.

IMDB Label Conversion: IMDB binary-class labels are converted to a rating score by label assignment rules — a negative label corresponds to random selection from $[1, 1.5, 2, 2.5]$ and a positive label from $[3, 3.5, 4, 4.5]$.

PeerRead Label: In the PeerRead training set, the ultimate score for rejection or acceptance of a paper is based on more than ten individual aspects such as originality and clarity, but often fewer than 5 aspect scores are available. Scores from other annotated aspects are averaged to fill in these missing aspects, introducing bias.

A.2 General STS with Heterogeneous Labels

We investigate the performance of our method in combining two heterogeneously-labelled datasets (SICK-R and STS-B), expanding the training set size but introducing noise.

STS-B is labelled in the range $[0, 5]$ in accordance with the standard STS formulation, while SICK-R is annotated in the range $[1, 5]$ with an emphasis on semantic relatedness rather than semantic similarity, leading to label misalignment in both label semantics and range between the two datasets. For example, completely irrelevant cases are scored 1.0 in SICK-R but 0 in STS-B, and for:

> **S1:** *A brown dog is attacking another animal in front of the man in pants.*
> **S2:** *Two dogs are fighting.*

the gold score is 3.5 in SICK-R, but for STS-B the score would be in $[1, 2]$.

Two alternatives exist to incorporate SICK-R into STS-B training: (1) fine-tuning jointly over combined training sets; and (2) fine-tuning first on STS-B, then on SICK-R. We investigate both. Consistent with the findings of Section 4, we pre-fine-tune over "STS-B", with training epochs=3, lr=2e-5, and without early stopping. We experiment with the full SICK-R training set ("SICK-R-full") and a subsample of 3000 instances ("SICK-R-3000").

Train Data	Manner	Discard	r	ρ	loss
STS-B	NA	NA	0.900	0.896	0.444
STS-B + SICK-R-3000	joint	No	0.900	0.896	0.440
STS-B + SICK-R-3000	joint	Yes	0.901	0.896	0.438
STS-B + SICK-R-3000	separate	No	0.874	0.881	1.515
STS-B + SICK-R-3000	separate	Yes	0.888	0.890	1.241
STS-B + SICK-R-full	joint	No	0.901	0.897	0.445
STS-B + SICK-R-full	joint	Yes	**0.903**	**0.898**	**0.430**
STS-B + SICK-R-full	separate	No	0.886	0.882	1.402
STS-B + SICK-R-full	separate	Yes	0.889	0.889	1.167

Table 8: Averaged loss, r and ρ on STS-B validation set **w/o** noise discarding combining SICK-R-3000 (sampled 3000 instances from SICK-R train set) or SICK-R-full (full train set) by joint and separate fine-tuning.

As presented in Table 8, using noise detection and discarding noisy instances can improve the performance under both joint and separate fine-tuning. Overall, in this scenario, from a strong baseline, the improvement is modest even though the training data volume is doubled in the case of the full SICK-R training set. This is largely because the "clean" data from SICK-R for STS-B purposes is mostly distributed in the range of $[4, 5]$, i.e., highly related pairs are also highly similar in the meaning, but this is generally the range where STS predictions are reliable, based on STS-B training. Put differently, even though the noise filtering method was able to discard noisy labels, it was ineffectual due to a lack of clean instances in the critical range $[2, 4]$ where STS-B models perform poorly (Mahajan et al., 2020).

This provides a valuable insight: it is vital to integrate examples in label ranges where the model is deficient. Further, given our findings, only joint fine-tuning is used in Section 6.