# A Light Label Denoising Method with the Internal Data Guidance

Anonymous ACL submission

## Abstract

Samples with incorrect labels are common in datasets, even annotated by humans. Some approaches have been proposed to alleviate the negative impact of mislabeling on the training process by removing erroneous data or reducing their weights. Unlike previous works, this paper introduces a light yet effective denoising method based on the relationship between the samples within the dataset, namely internal guidance. We examine the method on five datasets with mainstream models. The results demonstrate that this light denoising approach can obtain consistent improvement for all the datasets and models. [1]

## 1 Introduction

Machine learning benefits from high-quality labeled data. However, datasets often contain samples with erroneous labels (Xiao et al., 2015; Li et al., 2017b; Northcutt et al., 2021a) regardless of whether they are labeled by humans or automatic techniques. Since noisy labels inevitably bring negative impacts, the topic of mitigating the influences of the erroneous labels has received much attention.

A series of denoising approaches have been proposed to deal with the noisy samples, such as removing or modifying some samples (Song et al., 2019; Lyu and Tsang, 2019), or assigning small weights to noisy labels (Patrini et al., 2017; Hendrycks et al., 2018). However, most of these methods are designed in a task-specific style, which might lead to poor generalization capabilities. Another way of denoising focuses on effective and robust representations (Ghosh and Lan, 2021; Ciortan et al., 2021; Fang et al., 2020; Wu et al., 2020). For example, contrastive learning leverages data augmentation to create similar samples that help models to obtain robust representations. However, most augmentations only pay attention to one sample and its variants but neglect the relations between different samples.

Chan et al. (2021) claim that samples in the same class are inherently similar and correlated, and there are apparent differences in samples of different classes. Intuitively, in most situations, two sentences with similar content should be classified into the same class in the text classification task. However, we find that similar samples with the same label are not always assigned to the same class by models correctly. This problem could be more serious when the training data faces a certain level of noise, since the models are only supervised by the labels. It naturally raises the question: in addition to the label supervision, can we seek the guidance from the relationship between the samples in the training process?

To offer the internal guidance, this paper firstly proposes a novel representation of texts using weighted contextual information, and then employs the similarity between texts to guide the training process. We conduct experiments on five text classification datasets with two widely-used models. Empirical results show that this light denoising approach can achieve better performance than the existing label denoising methods. After introducing our method, consistent improvements are observed for all datasets and models, especially the tasks without sufficient training data or the datasets with high levels of noise. It should be noted that in our method, both the representation and the guiding process are built upon the dataset itself, without involving any external resources or introducing any extra parameters. Therefore, the internal guidance is light, efficient, and can be easily generalized to other datasets or tasks.

## 2 Related Work

Erroneous labels exist in most datasets, whether they are labeled manually or by machines. A straightforward method is reweighting contribu-

[1] The source code will be released on https://github.com.

tions of samples in loss function from the training aspect (Liu and Tao, 2015; Wang et al., 2017, 2018). However, since these methods depend on the manual design of weighting functions, it is difficult to apply them to other models and datasets with different noisy rates. Another set of studies handle the problem by improving the quality of datasets. Shen and Sanghavi (2019) and Lyu and Tsang (2019) update model parameters by selecting high-confident samples. Chen et al. (2019) change the selection process by iteration with two models, while Nguyen et al. (2019) leverage self-ensembling. Northcutt et al. (2021b) propose confident learning (Cleanlab) to estimate the joint distribution between noisy labels and unknown labels, and prune noisy data with probabilistic thresholds. However, these methods could possibly waste part of the data, since some correct but complex samples might be removed. Another concern is time-consuming because deleting samples needs to train several models or train one model for several times.

Contrastive learning could help models learn generalized and robust representations which could alleviate the negative influences of noisy data to some extent. One of the most important component in contrastive learning is data augmentation which generates positive and negative samples for pre-training. Different from computer vision (Chen et al., 2020), data augmentation in NLP focuses on text modification, e.g., back-translation (Fang et al., 2020), word/span deletion (Wu et al., 2020), and embedding dropout (Yan et al., 2021; Gao et al., 2021). Usually these methods require external corpora, but it is not easy to obtain suitable corpora for some downstream tasks. Meanwhile, contrastive learning could not model the relations between different samples since only a sample and its variants are considered similar in data augmentation.

# 3 Method

To illustrate the proposed label denoising method, we take text classification task as an example, since it is a classical task in NLP, and erroneous labels are common in the datasets. In text classification, a dataset with $n$ samples can be denoted as $D = \{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}, y_i \in \{c_1, c_2, ..., c_m\}$, where $x_i$ is the $i$-th input text, $y_i$ is its label and the dataset contains $m$ categories.

## 3.1 Contextual Representation

Text representation is crucial for modeling the similarities between samples. There are two typical ways to compute text similarity in NLP. One is based on symbolic representations, e.g. edit distance (Levenshtein et al., 1966), Jaccard similarity coefficient (JSC) (Jain et al., 2017), and Earth mover's distance (EMD) (Rubner et al., 2000). The other is to represent the texts in dense vectors (Mikolov et al., 2013; Devlin et al., 2019), and then obtain the vector similarities. However, the first method rely too much on the token repetition, and the second method requires representations pre-trained from the external corpus, which might miss the in-domain information within the dataset. Thus, we propose a new contextual representation based on Positive Pointwise Mutual Information (PPMI).

Firstly, we count the number of co-occurrences of words in a dataset with a sliding window. The co-occurrence matrix of words can be represented by $C$, where $C_{w_i w_j}$ is the number of a context word $w_j$ appears as a neighbor of a center word $w_i$. Then we calculate the PPMI matrix $E$ of the $C$:

$$E_{ij} = max(log \frac{P(w_i, w_j)}{P(w_i) \cdot P(w_j)}, 0) \quad (1)$$

where $P(w_i)$, $P(w_j)$ and $P(w_i, w_j)$ are the probability of $w_i$, $w_j$ and their co-occurrence in the dataset respectively. Lastly, the vector of $E_{w_i}$ is the representation of word $w_i$.

## 3.2 Word Weight

Since different words contribute to the meaning of text differently, we are more interested in the words that are helpful for classification than trivial words that appear in most of the samples, such as *a*, *the* and *of*. Different from Wang and Manning (2012) and Li et al. (2017a), we propose a variant to calculate the weight of word $w_i$ in a global view:

$$q_{w_i} = log \frac{(p_c^{w_i} + \alpha)/||p_c||_1}{(p_{\tilde{c}}^{w_i} + \alpha)/||p_{\tilde{c}}||_1} \quad (2)$$

where $c$ is the class that word $w_i$ has the highest frequency, $p_c^{w_i}$ is the count of samples that contain word $w_i$ in class $c$, $p_{\tilde{c}}^{w_i}$ is the count of samples that contain word $w_i$ in other classes, $||p_c||_1$ is the number of samples in class $c$ and $\alpha$ is a small value for smoothing, e.g., 0.1. Since some rare words only appear several times in the all dataset, they could not reflect the distribution precisely. Therefore, according to Zipf's law, we set the weight to 0 when the rank of a word is multiplied by its count beyond the mean plus/minus one standard deviation.

## 3.3 Guiding the Training

Given two pieces of text $a$ with $d$ words and $b$ with $e$ words, the similarity is

$$T_{sim}(a,b) = cosine(\sum_{i=1}^{d} q_{w_i} E_i, \sum_{j=1}^{e} q_{w_j} E_j) \quad (3)$$

Obviously, $T_{sim}(a,b)$ is always greater than zero.

In a classification task with $m$ categories, for the $i$-th sample, a model predicts a probability distribution of labels which can be denoted as $l_i = [l_{i1}, l_{i2}, ..., l_{ik}, ..., l_{im}]$, where $l_{ik} > 0$ and $\sum_{k=1}^{m} l_{ik} = 1$. The predicted similarity of two samples $a$ and $b$ is calculated by

$$L_{sim}(a,b) = cosine(l_a, l_b) \quad (4)$$

In the training phase, the loss function of a batch with $s$ samples is

$$loss = \sum_{k=1}^{s} CE(y_k, l_k) + \sum_{i=1}^{s}\sum_{j=1}^{s} \gamma (T_{sim}(i,j) - L_{sim}(i,j))^2$$
$$where\ \gamma = \begin{cases} 1, \text{if } T_{sim}(i,j) > \beta \\ 0, \text{otherwise} \end{cases} \quad (5)$$

where $CE$ means cross entropy and $\gamma$ is a factor to make similar samples prominent, i.e., the similarity of two samples will be set to zero when $T_{sim}(i,j) \leq \beta$ and $\beta$ is a small value.

# 4 Experiments

## 4.1 Datasets and Models

Five widely used text classification benchmarks are selected for our experiments. **MR** A movie review dataset (Pang and Lee, 2005) contains two categories. **SST-1** Stanford Sentiment Treebank (Socher et al., 2013) contains five emotion categories. **SST-2** A dataset removes the neutral label from SST-1, only retaining positive and negative emotions. **CR** A customer review dataset (Hu and Liu, 2004) contains two categories. **Subj** Subjectivity dataset (Pang and Lee, 2004) consists of the sentences with subjective or objective labels.

Since the approach is model-independent, we use two popular models with different architectures. **TextCNN** A simple yet effective model is widely used in text classification (Kim, 2014). **BERT** A pre-trained language model with an excellent performance in text classification (Devlin et al., 2019).

## 4.2 Experimental Setup

For TextCNN, we use the same hyperparameters and settings as Kim (2014). For BERT, we use $[CLS]$ for classification following Devlin et al. (2019), of which dropout rate is set to 0.1 and optimizer is Adam with learning rate of $2e^{-5}$. $\beta$ is set to 0.03, 0.05, 0.05, 0.1 and 0.2 for 0%, 10%, 20%, 30% and 40% noisy rate respectively.

Standard test sets are used in SST-1 and SST-2. 10-fold cross-validation is used for other datasets. Each experiment is repeated five times with different random seeds. We finally report the mean of them. All data pre-processing follow Kim (2014).

To conveniently observe the denoising performance of models, we add random noise to clean datasets. A label $y_i$ is randomly replaced by a label $\hat{y}_i$ with a probability $p_{noise}$ , where $\hat{y}_i \in \{c_1, c_2, ..., c_m\}$ and $\hat{y}_i \neq y_i$. Noisy labels are only added to the training set and validation set. The test set is clean and used to evaluate models.

## 4.3 Results and Analysis

As shown in Table 1, models do not benefit from Cleanlab in most of the experiments. One possible reason is that these datasets are too small for Cleanlab to delete samples correctly. Actually, Northcutt et al. (2021b) test it on one million samples, and they point out that the larger a dataset is, the more precisely Cleanlab estimates the probability of wrong labels. R-Drop leverages data augmentation to enhance representations, while it cannot deal with noisy labels as it only maximizes the KL-divergence of a sample and its variants rather than models the distribution of the whole dataset. In contrast, our method overcomes these drawbacks, which focuses on the intrinsic relation of data and does not need to remove any data. Thus it can perform well and stably even with small datasets.

LLD achieves consistent improvement on all datasets and gains more when the noisy rate increases. For TextCNN, the margin is notable whether in small datasets or in large datasets. Although BERT is pre-trained on a huge corpus, LLD also works when the size of data in downstream tasks is small. It should be noted that LLD obtains little advantage with BERT in large datasets (SST1/2) since BERT could be finetuned to acquire a good ability of generalization if there is sufficient data. Meanwhile, we find that pre-trained embeddings gain little improvement, as the information in PPMI totally comes from datasets which has less noise than embeddings pre-trianed with general corpus in similarity computation, which is similar with the phenomenon observed by Roberts (2016).

To compare the similarity metric with those based on explicit representation, e.g., edit distance, JSC and EMD, we compute the coverage of them by averaging the similarity of each sample with others, as shown in Table 2. Considering two samples may not be similar when their similarity is

| dataset | size | model | TextCNN | | | | | BERT | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | $p_{noise}$ | 0% | 10% | 20% | 30% | 40% | 0% | 10% | 20% | 30% | 40% |
| SST1 | 168672 | Baseline | 48.61 | 46.73 | 45.19 | 43.95 | 43.46 | 55.82 | 55.68 | 54.65 | 54.29 | 52.93 |
| | | Cleanlab | 48.17 | 45.24 | 42.80 | 43.95 | 42.23 | 55.20 | 53.89 | 53.79 | 51.98 | 51.78 |
| | | R-Drop | - | - | - | - | - | 55.82 | 54.85 | 53.54 | 53.33 | 52.35 |
| | | LLD-DW | 49.09 | 47.54 | 46.14 | 45.10 | 42.50 | 56.04 | 54.91 | 53.71 | 53.30 | 51.71 |
| | | LLD | **49.57** | 47.53 | **47.11** | **46.04** | **45.38** | **56.60** | 55.23 | 54.45 | 53.8 | 52.16 |
| SST2 | 79654 | Baseline | 87.89 | 86.43 | 85.91 | 83.71 | 76.45 | 93.06 | 92.98 | 91.46 | 89.7 | 78.72 |
| | | Cleanlab | 86.56 | 84.97 | 82.35 | 82.10 | 67.98 | 92.38 | 91.14 | 90.83 | 89.02 | 83.09 |
| | | R-Drop | - | - | - | - | - | 93.30 | 92.6 | 91.23 | 89.75 | 82.47 |
| | | LLD-DW | 87.60 | 86.27 | 85.32 | 82.74 | 74.14 | 93.57 | 92.67 | 90.55 | 88.30 | 75.94 |
| | | LLD | **88.22** | **86.72** | **86.3** | 83.63 | **79.19** | 93.19 | 92.67 | **92.05** | 89.59 | 80.51 |
| MR | 10662 | Baseline | 81.34 | 79.39 | 78.08 | 74.79 | 67.49 | 86.69 | 85.27 | 83.91 | 79.92 | 70.58 |
| | | Cleanlab | 79.19 | 77.30 | 75.84 | 72.67 | 64.98 | 84.63 | 83.63 | 77.22 | 74.28 | 61.81 |
| | | R-Drop | - | - | - | - | - | 87.06 | 85.42 | 83.63 | 78.99 | 66.89 |
| | | LLD-DW | 81.29 | 79.54 | 78.23 | 75.47 | 69.27 | 86.35 | 85.35 | 84.32 | 75.58 | 67.49 |
| | | LLD | 81.28 | **79.76** | **78.39** | **75.9** | **70.25** | 86.87 | **85.65** | **84.38** | **82.22** | **72.48** |
| CR | 3773 | Baseline | 84.17 | 82.54 | 79.18 | 74.52 | 65.85 | 90.96 | 88.88 | 87.95 | 83.24 | 71.46 |
| | | Cleanlab | 81.09 | 78.63 | 75.38 | 70.07 | 62.07 | 79.53 | 74.76 | 72.59 | 65.88 | 62.10 |
| | | R-Drop | - | - | - | - | - | 90.85 | 88.12 | 84.80 | 76.9 | 64.57 |
| | | LLD-DW | 84.12 | 82.27 | 79.81 | 76.08 | 67.66 | 90.16 | 88.97 | 86.55 | 81.65 | 63.11 |
| | | LLD | 83.91 | 82.29 | 79.77 | 75.86 | 67.51 | **91.04** | **89.59** | **88.02** | **85.00** | **73.33** |
| Subj | 10000 | Baseline | 93.17 | 91.46 | 90.37 | 88.35 | 82.13 | 96.59 | 95.50 | 94.93 | 93.33 | 88.98 |
| | | Cleanlab | 91.65 | 90.19 | 88.74 | 86.70 | 80.97 | 95.25 | 94.04 | 93.46 | 89.59 | 78.79 |
| | | R-Drop | - | - | - | - | - | 96.09 | 95.20 | 94.09 | 92.27 | 88.02 |
| | | LLD-DW | 93.08 | 91.70 | 90.72 | 89.28 | 84.88 | 96.27 | 95.79 | 94.72 | 94.12 | 89.31 |
| | | LLD | 92.79 | 91.67 | 90.68 | **89.33** | **86.22** | 96.44 | 95.64 | **95.06** | 93.96 | **91.67** |
| Avg. | - | Baseline | 79.04 | 77.31 | 75.75 | 73.06 | 67.08 | 84.62 | 83.66 | 82.58 | 80.10 | 72.53 |
| | | Cleanlab | 77.33 | 75.27 | 73.02 | 71.10 | 63.65 | 81.40 | 79.49 | 77.58 | 74.15 | 67.51 |
| | | R-Drop | - | - | - | - | - | 84.62 | 83.24 | 81.46 | 78.25 | 70.86 |
| | | LLD-DW | 79.04 | 77.47 | 76.04 | 73.73 | 67.69 | 84.48 | 83.54 | 81.97 | 78.59 | 69.51 |
| | | LLD | **79.15** | **77.59** | **76.45** | **74.15** | **69.71** | **84.83** | **83.76** | **82.79** | **80.91** | **74.03** |

Table 1: Accuracy with different noisy rates. Each result is the mean of five runs. R-Drop is proposed by Liang et al. (2021) which is based on contrastive learning without external corpus. Since scores are far lower than baselines, approximately 50% of baselines, when R-Drop is applied to TextCNN, it is unnecessary to show them. [2]To compare our method with data modification, we follow the official codes of Cleanlab to select data (Northcutt et al., 2021b), where all CL methods are tested and C+NR is reported as the best. LLD is a light denoising method proposed in this paper. LLD-DW uses the same processes with LLD except for replacing PPMI embeddings with word2vec embeddings. SST training set consists of sentences and phrases according to Kim (2014).

| $\beta$ | 0 | 0.3 | 0.5 |
|---|---|---|---|
| explicit | 0.1988 | 0.0203 | 0.0201 |
| LLD | 0.5109 | 0.3231 | 0.2134 |

Table 2: In SST-1, coverage of none-zero similarity varies with different $beta$. Explicit method means that the similarity is calculated by word overlap.

slightly greater than zero, we use $\beta$ to filter confusing samples. The explicit method can hardly judge how similar two samples are because only 19% of sample pairs are given scores greater than zero, and most of them actually are near zero when we filter them by a small $\beta$. However, scores of our method could cover a large part of data, which helps models classify similar samples into the same category which are disturbed by noise.

[2]We use the same setting with Liang et al. (2021) and get the similar result in SST2 with 0% $p_{noise}$. However, R-Drop does not have the ability to overcome the noise in these datasets, especially applied to TextCNN, although they claim R-Drop could help models to get better representations.

## 5 Conclusion

In this paper, we introduce a light yet effective denoising method with the guidance within the dataset. The aim is straightforward and intuitive, which is to make similar samples be classified in the same class. We firstly propose a new representation of texts based on weighted contextual information, and then leverage the similarity of texts to guide the training. Different from removing incorrect data or contrastive learning, this approach is built upon the dataset itself, without involving any external corpus and extra model parameters. The denoising method is model-independent, and we examine it in five classification tasks with two mainstream models. Empirical results illustrate that this light denoising approach can obtain consistent improvement on all datasets and models, especially with high levels of noise.

## References


Kwan Ho Ryan Chan, Yaodong Yu, Chong You, Haozhi Qi, John Wright, and Yi Ma. 2021. Redunet: A white-box deep network from the principle of maximizing rate reduction.

Pengfei Chen, Ben Ben Liao, Guangyong Chen, and Shengyu Zhang. 2019. Understanding and utilizing deep neural networks trained with noisy labels. In *International Conference on Machine Learning*, pages 1062–1070. PMLR.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR.

Madalina Ciortan, Romain Dupuis, and Thomas Peel. 2021. A framework using contrastive learning for classification with noisy labels. *Data*, 6(6):61.

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.

Hongchao Fang, Sicheng Wang, Meng Zhou, Jiayuan Ding, and Pengtao Xie. 2020. Cert: Contrastive self-supervised learning for language understanding. *arXiv preprint arXiv:2005.12766*.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821*.

Aritra Ghosh and Andrew Lan. 2021. Contrastive learning improves model robustness under label noise. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2703–2708.

Dan Hendrycks, Mantas Mazeika, Duncan Wilson, and Kevin Gimpel. 2018. Using trusted data to train deep networks on labels corrupted by severe noise. *Advances in Neural Information Processing Systems*, 31:10456–10465.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177.

Abhishek Jain, Aman Jain, Nihal Chauhan, Vikrant Singh, and Narina Thakur. 2017. Information retrieval using cosine and jaccard similarity measures in vector space model. *International Journal of Computer Applications*, 164(6):28–30.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.

Vladimir I Levenshtein et al. 1966. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710. Soviet Union.

Shen Li, Zhe Zhao, Tao Liu, Renfen Hu, and Xiaoyong Du. 2017a. Initializing convolutional filters with semantic features for text classification. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1884–1889, Copenhagen, Denmark. Association for Computational Linguistics.

Wen Li, Limin Wang, Wei Li, Eirikur Agustsson, and Luc Van Gool. 2017b. Webvision database: Visual learning and understanding from web data. *arXiv preprint arXiv:1708.02862*.

Xiaobo Liang, Lijun Wu, Juntao Li, Yue Wang, Qi Meng, Tao Qin, Wei Chen, Min Zhang, and Tie-Yan Liu. 2021. R-drop: Regularized dropout for neural networks. *arXiv preprint arXiv:2106.14448*.

Tongliang Liu and Dacheng Tao. 2015. Classification with noisy labels by importance reweighting. *IEEE Transactions on pattern analysis and machine intelligence*, 38(3):447–461.

Yueming Lyu and Ivor W Tsang. 2019. Curriculum loss: Robust learning and generalization against label corruption. In *International Conference on Learning Representations*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Duc Tam Nguyen, Chaithanya Kumar Mummadi, Thi Phuong Nhung Ngo, Thi Hoai Phuong Nguyen, Laura Beggel, and Thomas Brox. 2019. Self: Learning to filter noisy labels with self-ensembling. In *International Conference on Learning Representations*.

Curtis G Northcutt, Anish Athalye, and Jonas Mueller. 2021a. Pervasive label errors in test sets destabilize machine learning benchmarks. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*.

Curtis G. Northcutt, Lu Jiang, and Isaac L. Chuang. 2021b. Confident learning: Estimating uncertainty in dataset labels. *Journal of Artificial Intelligence Research (JAIR)*, 70:1373–1411.

Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd Annual Meeting of the Association*

*for Computational Linguistics (ACL-04)*, pages 271–278, Barcelona, Spain.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 115–124, Ann Arbor, Michigan. Association for Computational Linguistics.

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 *CVPR*.

Kirk Roberts. 2016. Assessing the corpus size vs. similarity trade-off for word embeddings in clinical nlp. In *Proceedings of the Clinical Natural Language Processing Workshop (ClinicalNLP)*, pages 54–63.

Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas. 2000. The earth mover's distance as a metric for image retrieval. *International journal of computer vision*, 40(2):99–121.

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.

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.

Hwanjun Song, Minseok Kim, and Jae-Gil Lee. 2019. SELFIE: Refurbishing unclean samples for robust deep learning. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5907–5915. PMLR.

Ruxin Wang, Tongliang Liu, and Dacheng Tao. 2017. Multiclass learning with partially corrupted labels. *IEEE transactions on neural networks and learning systems*, 29(6):2568–2580.

Sida Wang and Christopher Manning. 2012. Baselines and bigrams: Simple, good sentiment and topic classification. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 90–94, Jeju Island, Korea. Association for Computational Linguistics.

Yisen Wang, Weiyang Liu, Xingjun Ma, James Bailey, Hongyuan Zha, Le Song, and Shu-Tao Xia. 2018. Iterative learning with open-set noisy labels. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8688–8696.

Zhuofeng Wu, Sinong Wang, Jiatao Gu, Madian Khabsa, Fei Sun, and Hao Ma. 2020. Clear: Contrastive learning for sentence representation. *arXiv preprint arXiv:2012.15466*.

Tong Xiao, Tian Xia, Yi Yang, Chang Huang, and Xiaogang Wang. 2015. Learning from massive noisy labeled data for image classification. In *CVPR*.

Yuanmeng Yan, Rumei Li, Sirui Wang, Fuzheng Zhang, Wei Wu, and Weiran Xu. 2021. ConSERT: A contrastive framework for self-supervised sentence representation transfer. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5065–5075, Online. Association for Computational Linguistics.