Clean or Annotate: How to Spend a Limited Data Collection Budget

Derek Chen
ASAPP, New York, NY
dchen@asapp.com

Zhou Yu
Columbia University, NY
zy2461@columbia.edu

Samuel R. Bowman
New York University, NY
bowman@nyu.edu

Abstract

Crowdsourcing platforms are often used to collect datasets for training machine learning models, despite higher levels of inaccurate labeling compared to expert labeling. There are two common strategies to manage the impact of such noise: The first involves aggregating redundant annotations, but comes at the expense of labeling substantially fewer examples. Secondly, prior works have also considered using the entire annotation budget to label as many examples as possible and subsequently apply denoising algorithms to implicitly clean the dataset. We find a middle ground and propose an approach which reserves a fraction of annotations to *explicitly* clean up highly probable error samples to optimize the annotation process. In particular, we allocate a large portion of the labeling budget to form an initial dataset used to train a model. This model is then used to identify specific examples that appear most likely to be incorrect, which we spend the remaining budget to relabel. Experiments across three model variations and four natural language processing tasks show our approach outperforms or matches both label aggregation and advanced denoising methods designed to handle noisy labels when allocated the same finite annotation budget.

1 Introduction

Modern machine learning often depends on heavy data annotation efforts. To keep costs in check while maintaining speed and scalability, many people turn to non-specialist crowd-workers through platforms like Mechanical Turk. Although crowdsourcing reduces costs to a reasonable level, it also tends to produce substantially higher error rates compared with expert labeling. The classic approach for improving reliability in classification tasks is to perform redundant annotations which are later aggregated using a majority vote to form a single gold label (Snow et al., 2008; Sap et al., 2019a; Potts et al., 2021; Sap et al., 2019b). This solution is easy to understand and implement, but comes at the expense of severely reducing the number of labeled examples available for training.

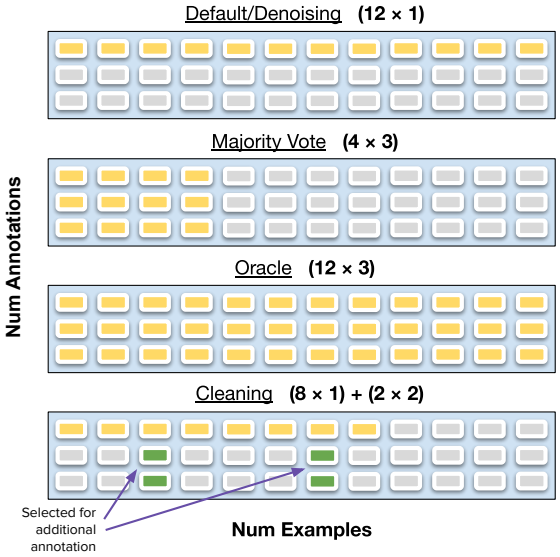

Figure 1: Data cleaning reserves a small portion of the annotation budget for targeted relabeling of examples that are identified as especially likely to be noisy. In contrast, the default and denoising methods spend the entire budget upfront, yielding lower quality data.

As an alternative, researchers have made great strides in designing automatic label cleaning methods, noise-insensitive training schemes and other mechanisms to work with noisy data (Sukhbaatar et al., 2015; Han et al., 2018; Tanaka et al., 2018). For example, some methods learn a noise transition matrix for reweighting the label (Dawid and Skene, 1979; Goldberger and Ben-Reuven, 2017), while others modify the loss (Ghosh et al., 2017; Patrini et al., 2017). Another set of options generate cleaned examples from mislabeled ones through semi-supervised pseudo-labeling (Jiang et al., 2018; Li et al., 2020). However, empirically getting many of these techniques to work well in practice is often a struggle due to the difficulty of training extra model components.

Proceedings of the Third Workshop on Deep Learning for Low-Resource Natural Language Processing, pages 152 - 168

We avoid the complexity of repairing or reweighting the labels of existing annotations by instead obtaining wholly new annotations from crowdworkers for a selected subset of samples. In doing so, our proposed methods require no extra model parameters to train, yet still retains the benefits of high label quality. Concretely, we start by allocating a large portion of the labeling budget to obtain an initial training dataset. The examples in this dataset are annotated in a single pass, and we would expect some percentage of them to be incorrectly labeled. However, enough of the labels should be correct to train a reasonable base model. Next, we take advantage of the recently trained model to identify incorrectly labeled examples, and then spend the remaining budget to relabel those examples. Finally, we train a new model using the original data combined with the cleaned data.

The key ingredient of our method is a function for selecting which examples to re-annotate. We consider multiple approaches for identifying candidates for relabeling, none of which have been applied before to denoising data within NLP settings. In all cases, relabeling the target examples relies on neither training any extra model components nor on tuning sensitive hyper-parameters. By using the existing annotation pipeline, the implementation becomes relatively trivial.

To test the generalizability of our method, we compare against multiple baselines on four tasks spanning multiple natural language formats. This departs from previous studies on human labeling in NLP, which focus exclusively on text classification (Wang et al., 2019; Jindal et al., 2019; Tayal et al., 2020). The control baseline and denoising baselines perform a single annotation per example. The majority vote baseline triples the annotations per example, but consequently is trained on only one third the number of examples to meet the annotation budget. We lastly include an oracle baseline that lifts the restriction on a fixed budget and instead uses all available annotations. We test across three model types, ranging from small ones taking minutes to train up to large transformer models which require a week to reach convergence. We find that under the same fixed annotation budget, cleaning methods match or surpass all baselines.

In summary, our contributions include:

1. We examine an alternative direction to learning with noisy labels that appear when data is collected under low-resource settings.
2. We build four versions of our approach that vary in how they target examples to relabel.
3. We compare against a number of baselines, many of which have never been implemented before in the natural language setting.

Overall, our *Large Loss* method, which selects examples for relabeling by the size of their training loss, performs the best out of all variations we consider despite requiring no extra parameters to train.

2 Related Work

The standard method for learning in the presence of unreliable annotation is to perform redundant annotation, where each example is annotated multiple times and a simple majority vote determines the final label (Snow et al., 2004; Russakovsky et al., 2015; Bowman et al., 2015). While effective, this can be costly since it severely reduces the amount of data collected. To tackle this problem, researchers have developed several alternative methods for dealing with noisy data that can be broken down into three categories.

Denoising Techniques Noisy training examples can be thought of as the result of perturbing the true, underlying labels by some source of noise. One group of methods assume the source of noise is from confusing one label *class* for another, and is resolved by reverting the errors through a noise transition matrix (Sukhbaatar et al., 2015; Goldberger and Ben-Reuven, 2017). Other methods work under the assumption that labeling errors occur due to *annotator* biases (Raykar et al., 2009; Rodrigues and Pereira, 2018), such as non-expert labelers (Welinder et al., 2010; Guan et al., 2018) or spammers (Hovy et al., 2013; Khetan et al., 2018). Finally, some methods model the noise of each individual *example*, either through expectation-maximization (Dawid and Skene, 1979; Whitehill et al., 2009; Mnih and Hinton, 2012), or neural networks (Felt et al., 2016; Jindal et al., 2019).

Another set of methods modify the loss function to make the model more robust to noise (Patrini et al., 2017). For example, some methods add a regularization term (Tanno et al., 2019), while others bound the amount of loss contributed by individual training examples (Ghosh et al., 2017; Zhang and Sabuncu, 2018). The learning procedure can also be modified such that the importance of training examples is dynamically reweighted to prevent overfitting to noise (Jiang et al., 2018).

Pseudo-labeling represents a final set of methods that either devise new labels for noisy data (Reed et al., 2015; Tanaka et al., 2018) or generate wholly new training examples (Arazo et al., 2019; Li et al., 2020). Other approaches from this family use two distinct networks to produce examples for each other to learn from (Han et al., 2018; Yu et al., 2019).

Budget Constrained Data Collection Our work also falls under research studying how to maximize the benefit of labeled data given a fixed annotation budget. Khetan and Oh (2016) apply model-based EM to model annotator noise, allowing singly-labeled data to outperform multiply-labeled data when annotation quality goes above a certain threshold. Bai et al. (2021) show that similar trade-offs exist when performing domain adaptation on a constrained budget. Zhang et al. (2021) observe that difficult examples benefit from additional annotations, so optimal spending actually varies the amount of labels given to each example. Our approach actively targets examples for relabeling based on its likelihood of noise, whereas they randomly select examples for multi-labeling without considering its characteristics.

Human in the Loop Finally, our work is also related to data labeling with humans. Annotators can be assisted through iterative labeling where models suggest labels for each training example (Settles, 2011; Schulz et al., 2019), or through active learning where models suggest which examples to label (Settles and Craven, 2008; Ash et al., 2020). In both cases, forward facing decisions are made on incoming batches of *unlabeled* data. In contrast, our methods look back to previously collected data to select examples for *relabeling*. These activities are orthogonal to each other and can both be included when training a model. (See Appendix C)

Lastly, re-active learning from (Sheng et al., 2008; Lin et al., 2016) proposes to relabel examples based on their predicted impact by retraining a classifier from scratch for every iteration of annotation. Accordingly, their method is impractical when adapted to the large Transformer models studied in this paper[1]. Instead, we identify examples to relabel through much less computationally expensive means, making the process tractable for real-life deployment.

[1]Training a large language model (such as RoBERTa-Large) until convergence can easily take a day or longer. Doing so each time for 12k annotations would take 30+ years.

3 Methods Under Study

We study how to maximize model performance given a static data annotation budget. Concretely, we are given some model M for a target task, along with a budget as measured by B number of annotations, where each annotation allows us to apply a possibly noisy labeling function $f_r(x)$, where r is the number of redundant annotations applied to a single example. Annotating some set of unlabeled instances produces noisy examples $(X, f_r(X)) = (X, \tilde{Y})$. Our goal is to achieve the best score possible for some primary evaluation metric S on a given task by cleaning the noisy labels $\tilde{Y} \xrightarrow{clean} Y$. Afterwards, we train a model with the cleaned data and then test it on a separate test set. For all our experiments, we set $B = 12,000$ as the total annotation budget.

As a default setting, we start with a *Control* baseline which uses the entire budget to annotate 12k examples, once each $(n = 12,000; r = 1)$. To simulate a single annotation, we randomly sample a label from the set of labels offered for each example by the dataset. To obtain more accurate labels, people often perform multiple annotations on each example and use *Majority Vote* to aggregate the annotations. Accordingly, as a second baseline we annotate 4k examples three times each $(n = 4,000; r = 3)$, matching the same total budget as before. In the event of a tie, we randomly select one of the candidate labels. Finally, we also include an *Oracle* baseline which uses the gold label for 12k examples $(n = 12,000; r = 3|5)$. The gold label is either given by the dataset or generated by majority vote, where the label might result from aggregating five annotations rather than just three annotations.

3.1 Noise Correction Baselines

We consider four advanced baselines, all of which perform a single annotation per example $(n = 12,000, r = 1)$ as seen in Figure 1. (1) (Goldberger and Ben-Reuven, 2017) propose applying a noise *Adaptation* layer which models the error probability of label classes. This layer is initialized as an identity matrix, which biases the layer to act as if there is no confusion in the labels. This noise transition matrix is then learned as a non-linear layer on top of the baseline model M to denoise predictions. The layer is discarded during final inference since gold labels are used during test time and are assumed to no longer be noisy.

(2) The *Crowdlayer* also operates by modeling the error probability, but assumes the noise arises due to annotator error, so a noise transition matrix is created for each worker (Rodrigues and Pereira, 2018). Once again, this matrix is learned with gradient descent and removed for final inference. (3) The *Forward* correction method from (Patrini et al., 2017) adopts a loss correction approach which modifies the training objective. Given $-\log p(\hat{y} = \tilde{y}|x)$ as the original loss, Forward modifies this to become $-log \sum_{j=1}^{c} T_{ji} p(\hat{y} = y|x)$ where c is the number of classes being predicted, and both i and j are used to index the number of classes. Matrix T is represented as a neural network that is learned jointly during pre-training. (4) Lastly, the *Bootstrap* method proposed by (Reed et al., 2015) generates pseudo-labels by gradually interpolating the predicted label $\hat{y}$ with the given noisy label $\tilde{y}$. We apply their recommended *hard* bootstrap variant which uses the one-hot prediction for interpolation since this was shown to work better in their experiments.

3.2 Cleaning through Targeted Relabeling

Rather than maximizing the number of examples annotated given our budget, we propose reserving a portion of the budget for reannotating the labels most likely to be incorrect. Specifically, we start by annotating a large number of examples one time each using the majority of the budget $(n_a = 10,000; r = 1)$. We then pretrain a model M_1 using this noisy data, and observe either the model's training dynamics or output predictions to target examples for relabeling. Next, we use the remaining budget to annotate those examples two more times $(n_b = 1,000; r = 2)$, allowing us to obtain a majority vote on those examples. The final training set is formed by combining the 1k multiply-annotated examples with the remaining 9k singly-annotated examples. We wrap up by initializing a new model M_2 with the weights from M_1 and fine-tune it with the clean data until convergence. We experiment with four approaches for discovering the most probable noisy labels:

Area Under the Margin AUM identifies problematic labels by tracking the margin between the likelihood assigned to the target label class and the likelihood of the next highest class as training progresses (Pleiss et al., 2020). Intuitively, if the gap between these two likelihoods is large, then the model is confident of its argmax prediction, presumably because the training label is correct. On the other hand, if the gap between them is small, or even negative, then the model is uncertain of its prediction, presumably because the label is noisy. AUM averages the margins over all training epochs and targets the examples with the smallest margins for relabeling.

Cartography Dataset Cartography is a technique for mapping the training dynamics of a dataset to diagnose its issues (Swayamdipta et al., 2020). The intuition is largely the same as AUM, such that Cartography also chooses consistently low-confidence (ie. low probability) examples for relabeling. We take the suggestion from Section 5 of their paper to detect mislabeled examples by tracking the mean model probability of the true label across epochs. Note that unlike AUM, Cartography tracks the final model outputs after the softmax, rather than the logits before the softmax. These can lead to different rankings since Cartography does not take the other probabilities in the distribution into account.

Large Loss (Arpit et al., 2017) found that correctly labeled examples are easier for a model to learn, and thus incur a small loss during training, whereas incorrectly labeled examples produce a large loss. Inspired by this observation and other similar works (Jiang et al., 2018), the Large Loss method selects examples for cleaning by ranking the top n_b examples where the model achieves the largest loss during the optimal stopping point. The ideal stopping point is the moment after the model has learned to fit the clean data, but before it has started to memorize the noisy data (Zhang et al., 2017). We approximate this stopping point by performing early stopping during training when the progression of the development set fails to improve for three epochs in a row. We then use the earlier checkpoint for identifying errors.

Prototype We lastly consider identifying noisy labels as those which are farthest away compared to the other training data (Lee et al., 2018). More specifically, we use a pretrained model to map all training examples into the same embedding space. Then, we select the vectors for each label class to form clusters where the centroid of each cluster is the "prototype" (Snell et al., 2017). Finally, we define outliers as those far away from the centroid for their given class, as measured by Euclidean distance, which are then selected for cleaning.

4 Experiments

4.1 Datasets and Tasks

To test our proposal, we select datasets that span across four natural language processing tasks. We choose these datasets because they provide multiple labels per example, allowing us to simulate single- and multiple-annotation scenarios.

Offense The Social Bias Frames dataset collects instances of biases and implied stereotypes found in text (Sap et al., 2020). We extract just the label of whether a statement is offensive for binary classification.

NLI We adopt the MultiNLI dataset for natural language inference (Williams et al., 2018). The three possible label classes for each sentence pair are *entailment*, *contradiction*, and *neutral*.

Sentiment Our third task uses the first round of the DynaSent corpus for four-way sentiment analysis (Potts et al., 2021). The possible labels are *positive*, *negative*, *neutral*, and *mixed*.

QA Our final task is question answering with examples coming from the NewsQA dataset (Trischler et al., 2017). The input includes a premise taken from a news article, along with a query related to the topic. The target label consists of two indexes representing the start and end locations within the article that extract a span of text answering the query. Unlike the other tasks, the format for QA is span selection rather than classification. Due to this distinction, certain denoising methods that assume a fixed set of candidate labels are omitted from comparison.

4.2 Training Configuration

In our experiments, we fine-tune parameters during initial training with only six runs, which is composed of three learning rates and two levels of dropout at 0.1 and 0.05. Occasionally, when varying dropout had no effect, we consider doubling the batch size instead from 16 to 32. We found an appropriate range of learning rates by initially conducting some sanity checks on a sub-sample of development data for each task and model combination. Learning rates were chosen from the set of [1e-6, 3e-6, 1e-5, 3e-5, 1e-4]. When a technique contained method-specific variables, we defaulted to the suggestions offered in their respective papers. We do not expect any of the methods to be particularly sensitive to specific hyperparameters.

4.3 Model Variations

We select three models for comparison that represent strong options at their respective model sizes. We repeat the process of example identification and simulated re-annotation separately for each model. We use all models as a pre-trained encoders to embed the text inputs of the different tasks we study.

DeBERTa-XLarge is our large model, which contains 750 million parameters and currently is the state-of-the-art on many natural language understanding tasks (He et al., 2021). DistilRoBERTa represents a distilled version of RoBERTa-base (Liu et al., 2019). It contains 82 million parameters, compared to the 125 million parameters found in RoBERTa. Learning follows the distillation process set by DistillBERT where a student model is trained to match the soft target probabilities produced by the larger teacher model (Sanh et al., 2019). Fine-tuning DistilRoBERTa is approximately 60-70 times faster compared to fine-tuning DeBERTa-XLarge on the same task.

For the final model, we avoid using Transformers altogether and instead use the FastText bag-of-words encoder (Joulin et al., 2017). The FastText embeddings are left unchanged during training, so the only learned parameters are in the 2-layer MLP we use for producing the model's final output. The same output prediction setup is used for all models, with a 300-dimensional hidden state. Training the FastText models run roughly 100-120 faster compared to working with DeBERTa-XLarge.

5 Major Results

Table 1 displays results across all models types and tasks, with each row representing a different technique. All rows except the Oracle were trained using the same label budget of 12,000 annotations.[2] In some cases, a method may surpass the Oracle since we conducted limited hyperparameter tuning, but as expected, the Oracle model outperforms all other methods overall. Notably, the Control setting always beats the Majority setting. In fact, Majority is consistently the lowest-performing method on all models and tasks, showing that improved label quality is never quite enough to overcome the reduction in annotation quantity. Adaptation is the best among denoising methods, achieving the

[2]Our annotation amount is much less than total available data for a task so our results are not directly comparable to prior work. For example, DynaSent train set includes 94,459 examples and Social Bias Frames contains 43,448 examples.

Methods	FastT	DRoB	DeXL	Avg
Oracle	*78.0*	*81.8*	*86.2*	*82.0*
Control	77.0	81.4	86.0	81.5
Majority	76.2	80.4	84.5	80.4
Adaptation	**77.8**	81.5	**86.1**	**81.8**
Crowdlayer	77.1	81.4	85.4	81.3
Bootstrap	77.1	81.2	85.1	81.2
Forward	77.5	81.2	84.9	81.2
Large Loss	77.7	**81.6**	85.4	81.6
AUM	77.5	81.5	85.3	81.4
Cartography	77.3	81.2	85.0	81.2
Prototype	77.7	81.4	85.5	81.5

(a) Offensive Language Detection from SBF

Methods	FastT	DRoB	DeXL	Avg
Oracle	*40.7*	*49.7*	*88.3*	*59.6*
Control	40.1	48.5	87.4	58.7
Majority	38.5	46.2	86.1	56.9
Adaptation	40.6	**49.4**	87.8	**59.2**
Crowdlayer	40.2	48.7	87.4	58.7
Bootstrap	**40.8**	49.3	87.4	59.1
Forward	40.6	48.6	87.3	58.8
Large Loss	40.5	48.9	87.8	59.1
AUM	40.3	49.0	87.1	58.8
Cartography	40.1	48.1	87.0	58.4
Prototype	40.4	48.6	**88.0**	59.0

(b) Natural Language Inference from MNLI

Methods	FastT	DRoB	DeXL	Avg
Oracle	*55.5*	*57.3*	*73.2*	*62.0*
Control	54.0	57.2	72.7	61.3
Majority	52.4	55.8	71.2	59.8
Adaptation	53.8	56.8	72.6	61.1
Crowdlayer	53.9	57.2	72.7	61.2
Bootstrap	54.1	**57.4**	72.7	61.4
Forward	53.5	57.3	73.0	61.4
Large Loss	**55.6**	**57.4**	**73.1**	**62.0**
AUM	55.4	56.5	72.6	61.5
Cartography	55.0	56.6	72.0	61.2
Prototype	55.1	57.1	**73.1**	61.7

(c) Sentiment Analysis from DynaSent

Methods	FastT	DRoB	DeXL	Avg
Oracle	—	*7.94*	*52.3*	*30.1*
Control	—	6.90	50.3	28.6
Majority	—	5.89	47.9	26.9
Adaptation	—	—	—	—
Crowdlayer	—	—	—	—
Bootstrap	—	6.72	50.5	28.6
Forward	—	—	—	—
Large Loss	—	**6.95**	**51.5**	**29.2**
AUM	—	6.69	**51.5**	29.1
Cartography	—	6.24	51.0	28.6
Prototype	—	—	—	—

(d) Question Answering from NewsQA

Table 1: Aggregated results for all method and model combinations, averaged over three seeds. Model names are abbreviated for space: FastT is FastText, DRoB is DistilRoBERTa, and DeXL is DeBERTa-XLarge. Avg is the average across models for that method. FastText doesn't produce context-dependent representations, and so is not usable on the QA task.

strongest results in two out of four settings. Large Loss is the best among cleaning methods, with the highest scores in the remaining two tasks. Prototypical is also a strong runner-up. Large Loss is the best overall method due to its consistency since it never drops below second on all tasks.

Variance among the three seeds is fairly consistent for all models and methods within the same task. Specifically, the standard deviation for offense detection and NLI are both around 0.5, with sentiment analysis and QA around 1.5 and 4.5, respectively. We do not see any strong trends across tasks, nor any outliers for a specific method.

Breakdown by Task Table 1a contains the results for offense language detection, where we see that Large Loss and Adaptation are the only methods to overtake the Control. These two are also the best overall performers on natural language inference as seen in Table 1b. The cleaning methods really shine on sentiment analysis and question answering where even the worst cleaning method often tops the best denoising method. We hypothesize this happens because the denoising methods work best in simple classification tasks, which we further explore in the next section. A handful of results are not reported in Table 1d since they refer to methods that are designed exclusively for classification tasks, and cannot be directly transferred to span selection.

Breakdown by Model The larger models perform better than the smaller models in terms of downstream accuracy, but somewhat surprisingly, there does not seem to be any clear patterns in relation to the method. In other words, if a particular method performs well (poorly) with one model size, it tends to also do well (poorly) when

	Large	AUM	Cart	Proto
Large Loss	1.000	0.541	0.000	0.316
AUM	---	1.000	0.001	0.212
Cartography	---	---	1.000	0.025
Prototype	---	---	---	1.000

Table 2: Jaccard similarity for all pairs of targeted relabeling methods on the sentiment analysis task. Large, Cart and Proto are short for Large Loss, Cartography and Prototype, respectively. Results for other tasks available in Appendix A.

Methods	Offense	NLI	Sentiment	QA
Default	81.6	48.9	57.4	6.95
Random	80.9	48.0	55.8	6.41
Cross	81.7	48.4	57.3	6.56

Table 3: Ablation results that vary the method of identifying errors for relabeling. Default uses the same model for error selection and training.

applied to the other model sizes too. One possible exception to this is the Prototype method showing strong performance with DeBERTA-XLarge. This is possibly because a stronger model produces more valuable hidden state representations for determining outliers. Since method performance is largely independent of the model size, we use DistillRoBERTa as the encoder for simplicity in the upcoming analyses.

Ablation How can we be sure that the cleaning methods are actually exhibiting a small, but consistent gain over the baselines rather than just natural variation? Perhaps the scores are close simply because all the methods use the same amount of training data. If the cleaning methods are indeed adding value, then they should perform much better than random selection. To measure this, we replace the pre-trained DistilRoBERTa model with a uniform sampler to identify examples for cleaning.

Active learning has been shown to exhibit significant decrease when transferring across model types (Lowell et al., 2019). In contrast, we argue that our method is not active learning since it is not directly dependent on the specific abilities of the target model. To test this claim, we also conduct an additional ablation whereby we replace one model type for another. Namely, we use the DeBERTa-XLarge model to select examples for cleaning, then train on the DistilRoBERTa model.

The results in Table 3 show that randomly selecting data points to relabel indeed lowers the final performance by a noticeable amount. By comparison, cross training models leads to a negligible drop in performance. We believe this occurs because targeted relabeling produces clean data, and clean data is useful regardless of the situation.

6 Discussion and Analysis

To better understand how the proposed clean methods operate, we conduct additional analysis with the sentiment analysis task.

Methods	Precision	GoEmotions	Synthetic
Oracle	—	*55.8*	*57.9*
Control	—	54.8	56.6
Majority	—	53.0	55.2
Adaptation	—	54.8	56.5
Crowdlayer	—	54.9	56.4
Bootstrap	—	55.0	**57.0**
Forward	—	53.9	56.2
Large Loss	56.8	**55.2**	56.5
AUM	60.4	54.6	56.1
Cartography	19.0	54.3	56.4
Prototype	46.6	55.1	56.7

Table 4: This table contains results for the three different post-hoc analyses. Left column is precision of the model in identifying mislabeled examples. Right columns are results training on extended datasets. All scores are average of three seeds on DistillRoBERTa.

How well do clean methods select items? We compare the four proposed methods by first looking at the amount of overlap in the examples selected for relabeling. To calculate this, we gather all examples chosen for relabeling by their likelihood of annotation error. For a given pair of methods, we then find the size of their intersection and divide by the size of their union, which yields the Jaccard similarity. As shown in Table 2, AUM and Large Loss have high overlap meaning that they select similar examples for cleaning. We additionally calculate the precision of each method by counting the number of times a label targeted for relabeling did not match the oracle label, and therefore legitimately requires cleaning. Based on Table 4, we once again see reasonable performance for the Large Loss cleaning method.

Qualitative examples for sentiment analysis are displayed in Table 5, which were chosen as the most likely examples of label errors according to their respective methods. Large Loss consistently discovers 'neutral' labels that were mis-labeled as

Method	Input Text	Label
Large Loss	That's usually how it go goes.	MIXED
	I always order "to-go"	MIXED
	It's $15 bucks for a beer since I used a drink ticket	MIXED
	We usually frequent the settlers ridge location.	MIXED
	I went on June 4th around 10:30.	MIXED
AUM	So fine, no problem.	POSITIVE
	A sirloin hotdog wrapped in bacon.	NEUTRAL
	For many years, I have gone to the Pet Smart down the street.	NEUTRAL
	I was always so happy here when it was managed by Johnny.	NEUTRAL
	I ordered the pad Thai noodles, chicken chow mien and egg rolls.	POSITIVE
Cartography	The food and customer service was fantastic when you first opened	POSITIVE
	The servers were pleasant.	POSITIVE
	Our waiter was overly friendly and informational.	MIXED
	Family owned and operated these folks are killing it	POSITIVE
	I really thought the young folks behind the counter were outgoing and seemed to enjoy their jobs	POSITIVE
Prototype	This should be a fun family place!	NEGATIVE
	Hotel was awesome.	NEGATIVE
	Great service for many years on our cars, but always at an additional price.	NEUTRAL
	Salad was great but a bit small.	NEUTRAL
	We had to specify the order *multiple* times, but eventually when the food came it was actually pretty good.	NEUTRAL

Table 5: Sentiment Analysis examples each method identified as being most likely to be label errors.

'mixed', while Prototype also does a good job uncovering label errors, finding 'positive' examples mislabeled as 'negative'. Overall, we see that the best performing cleaning methods do seem to pick up on meaningful patterns.

How many examples should be cleaned? All cleaning experiments so far have been run with n_a = 10,000 examples with n_b = 1,000 samples chosen for relabeling. This is equivalent to using up $\lambda = \frac{5}{6}$ of the labeling budget upfront, with the remaining annotations saved for later. This λ ratio was chosen as a reasonable default, but can also be tuned to be higher or lower. Figure 2 shows the results of varying the λ parameter from a range of $\frac{1}{6}$ to $\frac{11}{12}$. Based on the results, choosing $\lambda = \frac{2}{3}$ would have actually been the best option. This translates to n_a = 8,000 examples with n_b = 2,000 of those examples selected for re-labeling. As a sanity check, we also try dropping the n_b cleaned examples when retraining, keeping only the noisy data. As seen in Figure 2, the performance decreases as expected.

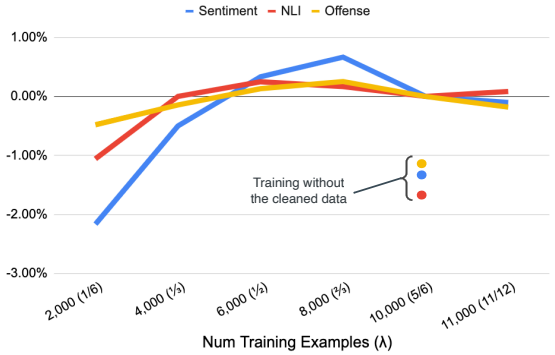

Figure 2: Varying the number of training examples changes the amount of budget remaining for cleaning. 10,000 examples is set as the default and the percent change is measured in comparison to this point.

What if we increase the number of classes? Based on the trends in the task breakdown of section 5, denoising methods seem to perform worse than explicit relabeling methods as the task gets harder. Most denoising methods may even become intractable for complex settings, such as those which require span selection. To test this hypothesis, we extend our setup to the GoEmotions dataset, where the goal of the task is to predict the emotion associated with a given utterance (Demszky et al., 2020). Whereas previous tasks dealt with 2-4 classes, the GoEmotions dataset requires a model to select from 27 granular emotions and a neutral option, for a total of 28 classes. Intuitively, we would expect the denoising methods to struggle since the pairwise interactions among classes has grown exponentially larger. The results in Table 4 reveal that Large Loss again outperforms all other methods in prediction accuracy. Notably, Adaptation in particular continues to exhibit lower than average scores compared to other methods. This supports our claim that relabeling methods are more robust as the number of classes grows.

What happens if noise is synthetically created? Many of the advanced denoising methods were originally tested on synthetically generated noise, whereas the noise in our datasets originates from noisy annotations, caused by the inherent ambiguity of natural language text (Pavlick and Kwiatkowski, 2019; Chen et al., 2020). Perhaps this partially explains how our proposed relabeling methods are able to outperform prior work. To study this further, we create a perturbed dataset starting from the gold DynaSent examples. Specifically, we randomly sample replacement labels according to a fabricated noise transition matrix, rather than sampling from the label distribution provided by annotators. (More details in Appendix D.) With noise coming from an explicit transition matrix, it might be easier for all models to pick up on this pattern.

The middle column of Table 4 shows that all eight cleaning methods perform on par with each other. When comparing the variance on this dataset with synthetic noise against the original DynaSent dataset with natural noise, the standard deviation drops from 0.34 down to 0.28, highlighting the uniformity in performance among the eight methods. The denoising methods work as intended on synthetic noise, but such assumptions may not hold on real data with more nuanced errors.

7 Conclusion

Noisy data is a common problem when annotating data under low resource settings. Performing redundant annotation on the same examples to mitigate noise leads to having even less data to work with, so we propose data cleaning instead through targeted relabeling. We apply our methods on multiple model sizes and NLP tasks of varying difficulty, which show that saving a portion of a labeling budget for re-annotation matches or outperforms other baselines despite requiring no extra parameters to train or hyper-parameters to tune. Intuitively, our best method can be summarized as double-checking the examples that the model gets wrong to see if it is actually an incorrect label causing problems.

Thus, to make the most out of the scarce labeled data available, we believe a best practice should include targeting examples for cleaning rather than spending the entire annotation budget upfront. Future work includes exploring more sophisticated techniques for identifying examples to relabel and understanding how such cleaning models perform on additional NLP tasks such as machine translation or dialogue state tracking, which have distinct output formats.

Acknowledgements

The authors would like to sincerely thank the reviewers for their attention to detail when reading through the paper. Their insightful questions and advice have noticeably improved the final manuscript. We would also like to thank ASAPP for sponsoring the costs of this project. Finally, we would like to acknowledge the helpful feedback from discussions with members of the Columbia Dialogue Lab.

References

Eric Arazo, Diego Ortego, Paul Albert, Noel E. O'Connor, and Kevin McGuinness. 2019. Unsupervised label noise modeling and loss correction. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 312–321. PMLR.

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, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 233–242. PMLR.

Jordan T. Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. 2020. Deep batch active learning by diverse, uncertain gradient lower bounds. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Fan Bai, Alan Ritter, and Wei Xu. 2021. Pre-train or annotate? domain adaptation with a constrained budget. *ArXiv*, abs/2109.04711.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009, Montreal, Quebec, Canada, June 14-18, 2009*, volume 382 of *ACM International Conference Proceeding Series*, pages 41–48. ACM.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 632–642. The Association for Computational Linguistics.

Tongfei Chen, Zhengping Jiang, Adam Poliak, Keisuke Sakaguchi, and Benjamin Van Durme. 2020. Uncertain natural language inference. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 8772–8779. Association for Computational Linguistics.

Alexander Philip Dawid and Allan M Skene. 1979. Maximum likelihood estimation of observer error-rates using the EM algorithm. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 28(1):20–28.

Dorottya Demszky, Dana Movshovitz-Attias, Jeongwoo Ko, Alan Cowen, Gaurav Nemade, and Sujith Ravi. 2020. GoEmotions: A Dataset of Fine-Grained Emotions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 4040–4054. Association for Computational Linguistics.

Paul Felt, Eric K. Ringger, and Kevin D. Seppi. 2016. Semantic annotation aggregation with conditional crowdsourcing models and word embeddings. In *COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan*, pages 1787–1796. ACL.

Aritra Ghosh, Himanshu Kumar, and P. S. Sastry. 2017. Robust loss functions under label noise for deep neural networks. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 1919–1925. AAAI Press.

Jacob Goldberger and Ehud Ben-Reuven. 2017. Training deep neural-networks using a noise adaptation layer. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Melody Y. Guan, Varun Gulshan, Andrew M. Dai, and Geoffrey E. Hinton. 2018. Who said what: Modeling individual labelers improves classification. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 3109–3118. AAAI Press.

Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor W. Tsang, and Masashi Sugiyama. 2018. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 8536–8546.

Steve Hanneke. 2014. Theory of disagreement-based active learning. *Found. Trends Mach. Learn.*, 7(2-3):131–309.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. DeBERTa: Decoding-enhanced BERT with disentangled attention. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Dirk Hovy, Taylor Berg-Kirkpatrick, Ashish Vaswani, and Eduard H. Hovy. 2013. Learning whom to trust with MACE. In *Human Language Technologies:*

Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 9-14, 2013, Westin Peachtree Plaza Hotel, Atlanta, Georgia, USA, pages 1120–1130. The Association for Computational Linguistics.

Lu Jiang, Deyu Meng, Shoou-I Yu, Zhen-Zhong Lan, Shiguang Shan, and Alexander G. Hauptmann. 2014. Self-paced learning with diversity. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 2078–2086.

Lu Jiang, Deyu Meng, Qian Zhao, Shiguang Shan, and Alexander G. Hauptmann. 2015. Self-paced curriculum learning. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA*, pages 2694–2700. AAAI Press.

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 *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 2309–2318. PMLR.

Ishan Jindal, Daniel Pressel, Brian Lester, and Matthew S. Nokleby. 2019. An effective label noise model for DNN text classification. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 3246–3256. Association for Computational Linguistics.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomás Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 2: Short Papers*, pages 427–431. Association for Computational Linguistics.

Ashish Khetan, Zachary C. Lipton, and Animashree Anandkumar. 2018. Learning from noisy singly-labeled data. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Ashish Khetan and Sewoong Oh. 2016. Achieving budget-optimality with adaptive schemes in crowdsourcing. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 4844–4852.

M. Pawan Kumar, Benjamin Packer, and Daphne Koller. 2010. Self-paced learning for latent variable models. In *Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010. Proceedings of a meeting held 6-9 December 2010, Vancouver, British Columbia, Canada*, pages 1189–1197. Curran Associates, Inc.

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 Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 5447–5456. Computer Vision Foundation / IEEE Computer Society.

Junnan Li, Richard Socher, and Steven C. H. Hoi. 2020. Dividemix: Learning with noisy labels as semi-supervised learning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Christopher H. Lin, Mausam, and Daniel S. Weld. 2016. Re-active learning: Active learning with relabeling. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, pages 1845–1852. AAAI Press.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *ArXiv*, abs/1907.11692.

David Lowell, Zachary C. Lipton, and Byron C. Wallace. 2019. Practical obstacles to deploying active learning. 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 2019, Hong Kong, China, November 3-7, 2019*, pages 21–30. Association for Computational Linguistics.

Volodymyr Mnih and Geoffrey E. Hinton. 2012. Learning to label aerial images from noisy data. In *Proceedings of the 29th International Conference on Machine Learning, ICML 2012, Edinburgh, Scotland, UK, June 26 - July 1, 2012*. icml.cc / Omnipress.

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 *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 2233–2241. IEEE Computer Society.

Ellie Pavlick and Tom Kwiatkowski. 2019. Inherent disagreements in human textual inferences. *Trans. Assoc. Comput. Linguistics*, 7:677–694.

Geoff Pleiss, Tianyi Zhang, Ethan R. Elenberg, and Kilian Q. Weinberger. 2020. Identifying mislabeled

data using the area under the margin ranking. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Christopher Potts, Zhengxuan Wu, Atticus Geiger, and Douwe Kiela. 2021. Dynasent: A dynamic benchmark for sentiment analysis. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 2388–2404. Association for Computational Linguistics.

Vikas C. Raykar, Shipeng Yu, Linda H. Zhao, Anna K. Jerebko, Charles Florin, Gerardo Hermosillo Valadez, Luca Bogoni, and Linda Moy. 2009. Supervised learning from multiple experts: whom to trust when everyone lies a bit. In *Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009, Montreal, Quebec, Canada, June 14-18, 2009*, volume 382 of *ACM International Conference Proceeding Series*, pages 889–896. ACM.

Scott E. Reed, Honglak Lee, Dragomir Anguelov, Christian Szegedy, Dumitru Erhan, and Andrew Rabinovich. 2015. Training deep neural networks on noisy labels with bootstrapping. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Workshop Track Proceedings*.

Filipe Rodrigues and Francisco C. Pereira. 2018. Deep learning from crowds. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 1611–1618. AAAI Press.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Li Fei-Fei. 2015. Imagenet large scale visual recognition challenge. *Int. Journal of Computer Vision*, 115(3):211–252.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108.

Maarten Sap, Ronan Le Bras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A. Smith, and Yejin Choi. 2019a. ATOMIC: an atlas of machine commonsense for if-then reasoning. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 3027–3035. AAAI Press.

Maarten Sap, Saadia Gabriel, Lianhui Qin, Dan Jurafsky, Noah A. Smith, and Yejin Choi. 2020. Social bias frames: Reasoning about social and power implications of language. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 5477–5490. Association for Computational Linguistics.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. 2019b. Social iqa: Commonsense reasoning about social interactions. 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 2019, Hong Kong, China, November 3-7, 2019*, pages 4462–4472. Association for Computational Linguistics.

Claudia Schulz, Christian M. Meyer, Jan Kiesewetter, Michael Sailer, Elisabeth Bauer, Martin R. Fischer, Frank Fischer, and Iryna Gurevych. 2019. Analysis of automatic annotation suggestions for hard discourse-level tasks in expert domains. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2761–2772, Florence, Italy. Association for Computational Linguistics.

Ozan Sener and Silvio Savarese. 2018. Active learning for convolutional neural networks: A core-set approach. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Burr Settles. 2011. Closing the loop: Fast, interactive semi-supervised annotation with queries on features and instances. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1467–1478, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Burr Settles and Mark Craven. 2008. An analysis of active learning strategies for sequence labeling tasks. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 1070–1079, Honolulu, Hawaii. Association for Computational Linguistics.

Victor S. Sheng, Foster J. Provost, and Panagiotis G. Ipeirotis. 2008. Get another label? improving data quality and data mining using multiple, noisy labelers. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, Nevada, USA, August 24-27, 2008*, pages 614–622. ACM.

Jake Snell, Kevin Swersky, and Richard S. Zemel. 2017. Prototypical networks for few-shot learning.

In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 4077–4087.

Rion Snow, Daniel Jurafsky, and Andrew Y. Ng. 2004. Learning syntactic patterns for automatic hypernym discovery. In *Advances in Neural Information Processing Systems 17 [Neural Information Processing Systems, NIPS 2004, December 13-18, 2004, Vancouver, British Columbia, Canada]*, pages 1297–1304.

Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Y. Ng. 2008. Cheap and fast - but is it good? evaluating non-expert annotations for natural language tasks. In *2008 Conference on Empirical Methods in Natural Language Processing, EMNLP 2008, Proceedings of the Conference, 25-27 October 2008, Honolulu, Hawaii, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 254–263. ACL.

Sainbayar Sukhbaatar, Joan Bruna, Manohar Paluri, Lubomir Bourdev, and Rob Fergus. 2015. Training convolutional neural networks with noisy labels. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Workshop Track Proceedings*.

Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A. Smith, and Yejin Choi. 2020. Dataset cartography: Mapping and diagnosing datasets with training dynamics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 9275–9293. Association for Computational Linguistics.

Daiki Tanaka, Daiki Ikami, Toshihiko Yamasaki, and Kiyoharu Aizawa. 2018. Joint optimization framework for learning with noisy labels. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 5552–5560. Computer Vision Foundation / IEEE Computer Society.

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, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 11244–11253. Computer Vision Foundation / IEEE.

Kshitij Tayal, Rahul Ghosh, and Vipin Kumar. 2020. Model-agnostic methods for text classification with inherent noise. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020 - Industry Track, Online, December 12, 2020*, pages 202–213. International Committee on Computational Linguistics.

Simon Tong and Daphne Koller. 2001. Support vector machine active learning with applications to text classification. *J. Mach. Learn. Res.*, 2:45–66.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. 2017. NewsQA: A machine comprehension dataset. In *Proceedings of the 2nd Workshop on Representation Learning for NLP, Rep4NLP at ACL 2017, Vancouver, Canada, August 3, 2017*, pages 191–200. Association for Computational Linguistics.

Hao Wang, Bing Liu, Chaozhuo Li, Yan Yang, and Tianrui Li. 2019. 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 2019, Hong Kong, China, November 3-7, 2019*, pages 6285–6291. Association for Computational Linguistics.

Peter Welinder, Steve Branson, Serge J. Belongie, and Pietro Perona. 2010. The multidimensional wisdom of crowds. In *Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010. Proceedings of a meeting held 6-9 December 2010, Vancouver, British Columbia, Canada*, pages 2424–2432. Curran Associates, Inc.

Jacob Whitehill, Paul Ruvolo, Tingfan Wu, Jacob Bergsma, and Javier R. Movellan. 2009. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *Advances in Neural Information Processing Systems 22: 23rd Annual Conference on Neural Information Processing Systems 2009. Proceedings of a meeting held 7-10 December 2009, Vancouver, British Columbia, Canada*, pages 2035–2043. Curran Associates, Inc.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

Xingrui Yu, Bo Han, Jiangchao Yao, Gang Niu, Ivor W. Tsang, and Masashi Sugiyama. 2019. How does disagreement help generalization against label corruption? In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 7164–7173. PMLR.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. 2017. Understanding deep learning requires rethinking generalization.

In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Shujian Zhang, Chengyue Gong, and Eunsol Choi. 2021. Learning with different amounts of annotation: From zero to many labels. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 7620–7632. Association for Computational Linguistics.

Zhilu Zhang and Mert R. Sabuncu. 2018. Generalized cross entropy loss for training deep neural networks with noisy labels. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 8792–8802.

A Additional Quantitative Results

Looking at Figure 3, the similarity scores for offensive language detection and natural language inference largely match up with the scores found in sentiment analysis. In particular, Large Loss and AUM exhibit higher overlap with each other. Additionally, Prototype shows a medium overlap and Cartography shows no overlap at all with the other methods. We reach a similar conclusion that the Large Loss method is a reasonable technique.

B Additional Qualitative Examples

More examples can be found in Table 6 on the next page. We see that Large Loss is once again quite accurate in picking up labeling errors. Prototype for NLI does a great job at finding examples labeled as 'entailment' which should be something else. The hypotheses for all the selected examples contain negative sentiment, which may be located far away from the entailment examples in the embedding space. Cartography exhibits a pattern of always choosing examples labeled as 'contradiction'.

C Comparison to Learning Schemes

On the surface, targeting examples for relabeling contains may seem similar to active learning or curriculum learning. Although there are certainly some parallels between these techniques, these are fundamentally different learning paradigms.

Active learning methods typically choose new examples to label based on the uncertainty of the model (Tong and Koller, 2001; Hanneke, 2014) or on the diversity they can add to the existing distribution (Sener and Savarese, 2018; Ash et al., 2020). Although sample noise can also be measured through model uncertainty, denoising and active learning do not have the same goal. More specifically, the noise of a training example is related to how its label is somehow incorrect. Perhaps the start of a span was not properly selected or an example that should not be tagged was accidentally included. More simply, an example is mislabeled as class A, when in fact it belongs to class B. This situation is not possible with active learning because the examples in active learning do not have labels yet! The entire point of active learning is to choose which examples should be labeled next (Settles and Craven, 2008; Settles, 2011). Thus, when we try to identify examples for cleaning, we are *re*-labeling rather than labeling for the first time.

Curriculum learning also selects examples for training based on model uncertainty (Bengio et al., 2009) and diversity maximization (Jiang et al., 2014). It could be interpreted that easier examples are those that contain less noise, which would connect to our proposed process. However, traditional curriculum learning chooses these examples upfront rather than based on modeling dynamics (Jiang et al., 2015). Extensions have been made under the umbrella of self-paced curriculum learning whereby examples are chosen for a curriculum based on how they react to a model's behavior (Kumar et al., 2010). This is indeed similar to how we can choose to relabel examples based on the model loss. This aspect of relabeling though is the key distinction – we *act* on these examples in an attempt to denoise the dataset. On the other hand, self-paced learning simply feeds those same examples back into the model without any modification.

D Data Preprocessing

D.1 Synthetic Data Generation

The synthetic dataset is created by applying an explicit noise transition matrix with 20% noise. Since the original dataset contains four classes, we start with an empty 4x4 matrix. The labels should not be confused most of the time so we assign a likelihood of 0.8 across the diagonal of the matrix. Next, we randomly select another class for each row to receive 0.1 likelihood of confusion. This leaves 0.1 for each row to be divided between the two remaining classes, which receive 0.05 each. For each example in the oracle dataset, we use the original label to select a single row from the constructed noise transition matrix. Lastly, we are able to sample a new label according to the weights provided by this 4-D vector. In contrast, the original sampling procedure obtained its weights according to the normalized label distribution provided by the annotations.

D.2 GoEmotions Preprocessing

To prepare the GoEmotions dataset, we filter the raw data to include only examples that have at least three annotators and a clear majority vote (used for determining the gold label). We then cross-reference this against the proposed data splits offered by the authors which have high inter-annotator agreement. From this joint pool of examples, we sample 12k training examples to match the setting of all our other experiments. This results in

	Large	AUM	Cart	Proto
Large Loss	1.000	0.637	0.000	0.190
Area Margin	---	1.000	0.000	0.125
Cartography	---	---	1.000	0.166
Prototype	---	---	---	1.000

(a) Jaccard similarity on Social Bias Frames

	Large	AUM	Cart	Proto
Large Loss	1.000	0.545	0.000	0.191
Area Margin	---	1.000	0.000	0.202
Cartography	---	---	1.000	0.152
Prototype	---	---	---	1.000

(b) Jaccard similarity on MNLI dataset

Figure 3: Jaccard similarity overlap for all pairs of targeted relabeling methods on the offensive language detection task and the natural language inference task.

12000/2954/2946 examples for train, development and test splits respectively.

E Limitations

Our proposed methods are limited to studying noise which comes from human annotators acting in good faith. Other sources of label noise include errors which occur due to spammers, distant supervision (as commonly seen in Named Entity Recognition), and/or pseudo-labels from bootstrapping. Within interactive settings, such as for dialogue systems, models may also encounter noisy user inputs such as out-of-domain requests or ambiguous queries. Our methods would not work well in those regimes either.

Method	Premise	Hypothesis	Label
	Why shouldn't he be?	He doesn't actually want to be that way.	ENTAILMENT
	How do they feel about your being a Theater major?	They don't know you're a theater major, do they?	ENTAILMENT
	Defication of humankind as supreme.	Humankind is not supreme.	ENTAILMENT
Large Loss	These are artists who are either emerging as leaders in their fields or who have already become well known.	These artists are becoming well known in their fields.	CONTRADICTION
	As he stepped across the threshold, Tommy brought the picture down with terrific force on his head.	Tommy stepped across a threshold and put a picture down on his head.	CONTRADICTION
	And if, as ultimately happened, no settlement resulted, we could shrug our shoulders, say, 'Hey, we tried.'	Even if an agreement could not be reach we could say we tried.	ENTAILMENT
	Companies that were foreign had to accept Indian financial participation and management.	Foreign companies had to take Italian money	CONTRADICTION
AUM	... he's been tireless about pursuing both celebrity and the cause of popular history ever since.	He never wanted any attention and kept to himself all the time.	CONTRADICTION
	Two more weeks with my cute TV satellite dish have increased my appreciation of it.	My appreciation of my satellite dish has increased.	ENTAILMENT
	Each working group met several times to develop recommendations for ... legal services delivery system	Each working met more than once to discuss changes to the legal services delivery system.	ENTAILMENT
	A detailed English explanation of the plot is always provided, and wireless recorded commentary units ...	You'll have to figure the plot out on your own.	CONTRADICTION
	I just loved Cinderella . I also saw my sisters as the wicked stepsisters sometimes, and I was Cinderella ...	I really disliked Cinderella and could never relate to her.	CONTRADICTION
Cartography	The judge gave vent to a faint murmur of disapprobation and the prisoner in the dock leant forward angrily.	The prisoner in the dock remained still and and expressionless	CONTRADICTION
	Jon was about to require a lot from her.	Jon needed nothing to do with her.	CONTRADICTION
	I know you'll enjoy being a part of the Herron School of Art and Gallery.	You will detest the Herron School of Art and Gallery and have nothing to do with it	CONTRADICTION
	Why shouldn't he be?	He doesn't actually want to be that way.	ENTAILMENT
	I like this area a whole lot and it's, it's growing so much and I just want to be near my family ...	I really despise living in this location and would prefer to be farther away from my relatives.	ENTAILMENT
Prototype	The air is warm.	The arid air permeates the surrounding land.	ENTAILMENT
	Inside the Oval: White House Tapes From FDR to Clinton	No tapes were recorded in the white house	ENTAILMENT
	He became even more concerned as its route changed moving into another sector's airspace.	He wasn't worried at all for the plane	ENTAILMENT

Table 6: Natural language inference examples that each method identified as being most likely to be label errors. Sentences were truncated in some cases for brevity.