Detecting Label Errors by using Pre-Trained Language Models

Derek Chong*
Stanford University
derekch@stanford.edu

Jenny Hong*
Stanford University
jennyhong@cs.stanford.edu

Christopher D. Manning
Stanford University
manning@cs.stanford.edu

Abstract

We show that large pre-trained language models are inherently highly capable of identifying label errors in natural language datasets: simply examining out-of-sample data points in descending order of fine-tuned task loss significantly outperforms more complex error-detection mechanisms proposed in previous work. To this end, we contribute a novel method for introducing realistic, human-originated label noise into existing crowdsourced datasets such as SNLI and TweetNLP. We show that this noise has similar properties to real, hand-verified label errors, and is harder to detect than existing synthetic noise, creating challenges for model robustness. We argue that human-originated noise is a better standard for evaluation than synthetic noise. Finally, we use crowdsourced verification to evaluate the detection of real errors on IMDB, Amazon Reviews, and Recon, and confirm that pre-trained models perform at a 9–36% higher absolute Area Under the Precision-Recall Curve than existing models.

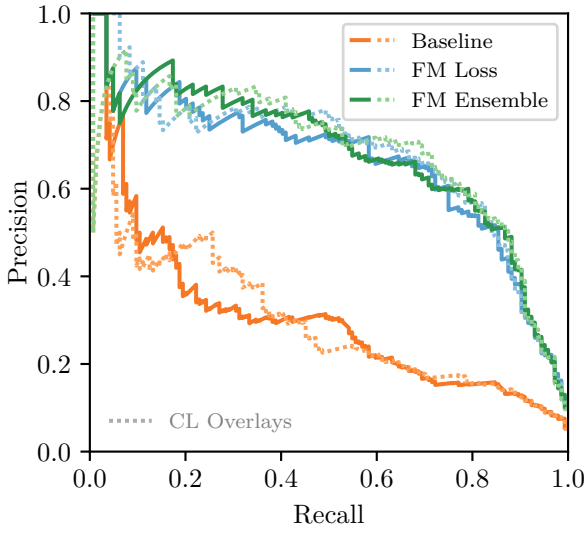

Figure 1: Precision-recall curves for label error detection: Large language models detect label errors with high precision, and far more effectively than a baseline word vector-based neural classifier. Overlaying a state-of-the-art model-agnostic error detection method, Confident Learning, results in little to no improvement (TweetNLP-5; §7).

1 Introduction

Improving model performance in the presence of label errors comprises an area of active research (Song et al., 2022). However, existing methods focus on label errors in training data. Although seldom acknowledged, evaluation label errors are at least as pernicious as training label errors: pervasive errors in commonly used NLP benchmarks have been found to destabilize model performance (Malik and Bhardwaj, 2011; Northcutt et al., 2021b). Such findings suggest that improving training methods does not preclude the need for improving the underlying data. We propose a simple method for using large, pre-trained language models (LLMs) to directly identify label errors for the purposes of correcting or removing them.

The majority of work in identifying label errors, and in data-centric artificial intelligence (DCAI) more broadly, focuses on image and healthcare data (DCAI Workshop, 2021). However, the success of the foundation model (FM) paradigm in applying pre-trained language models to a variety of NLP tasks (Bommasani et al., 2021; Reiss et al., 2020) suggests that FMs may be a powerful tool for detecting and correcting label errors in language datasets. Pre-training has been shown to imbue models with properties such as resistance to label errors, class imbalance (Karthik et al., 2021), out-of-distribution detection (Hendrycks et al., 2018), and confidence calibration (Desai and Durrett, 2020), while conferring ro-

*Equal contribution.

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 9074 - 9091
December 7-11, 2022

Dataset	Text	Label	Sentiment
IMDB	It is really unfortunate that a movie so well produced turns out to be such a disappointment. I thought this was full of (silly) cliches. It had all sorts of differences that it tried to tie together (not a bad thing in itself) but the result is at best awkward, but in fact ridiculous–too many clashes that wouldn't really happen. Then the end of the movie–the last 10 minutes–ruined all the rest. At first I thought Xavier was OK but with retrospect I think he was pretty bad. And that's all really too bad, because technically it was really good, and the soundtrack was great too. So the form was good, but the content pretty horrible.	~~Positive~~	Negative
IMDB	The ending made my heart jump up into my throat. I proceeded to leave the movie theater a little jittery. After all, it was nearly midnight. The movie was better than I expected. I don't know why it didn't last very long in the theaters or make as much money as anticipated. Definitely would recommend.	~~Negative~~	Positive
Amazon	The new design only has a thin layer of cellulose sponge material. It will not last as long. Already showing signs of wearing out. The picture does not represent the item received.	~~Neutral~~	Negative

Table 1: Organic label errors from sentiment datasets IMDB and Amazon, shown with the original dataset label. Each example was hypothesized by our model to be erroneous, and later verified by crowd workers.

bustness, generalization, and natural language understanding capabilities (Wang et al., 2018; Petroni et al., 2019). Our primary contribution is to show that simply verifying items in order of their out-of-sample loss on a foundation model improves precision by an absolute 15–28% and Area Under the Precision-Recall Curve (AUPR) by an absolute 9–36%.

Many methods for label error detection rely on artificially introduced label errors as ground truth for evaluating their methods. Northcutt et al. (2021a) develop a state-of-the-art model for identifying label errors, Confident Learning (CL), and use the better approach of crowdsourced human evaluations to determine the ground truth of label errors. We model our experiments on real data after their verification protocol, replicating this on real errors in IMDB (Maas et al., 2011), Amazon Reviews (McAuley et al., 2015), and Recon (Hong et al., 2021), with adaptations to mitigate annotator fraud (Kennedy et al., 2020).

In the process of assessing our results, we contribute a novel technique and protocol for introducing realistic, human-originated label noise into existing crowdsourced datasets, and apply it to two such datasets, TweetNLP (Gimpel et al., 2010) and SNLI (Bowman et al., 2015). We demonstrate that our technique better approximates *organic* (real, naturally occurring) label errors than existing methods. We provide evidence that this realism is essential to properly assessing model performance: even models that are robust to standard synthetic noising approaches show limited robustness to human-originated noise.[1]

2 Related Work

Learning with Noisy Labels (LNL) focuses on the model-training stage. Noise-robust approaches examine model enhancements such as the design of loss functions (Joulin et al., 2016; Amid et al., 2019; Liu and Guo, 2020; Ma et al., 2020), regularization (Azadi et al., 2015; Zhou and Chen, 2021), reweighting (Bar et al., 2021; Kumar and Amid, 2021), hard negative mining and contrastive learning (Zhang and Stratos, 2021). Noise-cleansing approaches aim to segregate clean data from noisy data in training, e.g. bagging and boosting (Wheway, 2000; Sluban et al., 2014), k-nearest neighbors (Delany et al., 2012), outlier detection (Gamberger et al., 2000; Thongkam et al., 2008), bootstrapping (Reed et al., 2014), and neural networks supervised directly on detecting an error, when such data exist (Jiang et al., 2018).

LNL methods have in most cases been evaluated using artificially-generated label noise. A typical evaluation of an LNL method uses a standard benchmark dataset, and programmatically corrupts training labels via one of three main noising schemes (Frenay and Verleysen, 2014; Algan and Ulusoy, 2020). *Uni-*

[1]Data noising library and evaluation data available at https://github.com/dcx/lnlfm.

form noise is most commonly used but unrealistic; deep neural networks have been found to perform well even when noised labels outnumber original labels at a ratio of 100 to 1 (Rolnick et al., 2017). *Class-dependent noise* randomly permutes labels based on a confusion matrix. However, research on annotator disagreement suggests that label errors tend to result from feature-based, not class-based ambiguity (Hendrycks et al., 2018). Training models to generate realistic *feature-based* or *instance-dependent noise* has recently emerged as an area of active research (Chen et al., 2021b; Xu et al., 2021; Dawson and Polikar, 2021). However, Algan and Ulusoy (2020) report that feature-dependent noise may bias benchmark performance toward similar models to the ones used to generate this noise.

The noising schemes above each fail in some way to simulate organic, naturally occurring label errors, which are estimated to occur in common benchmarks at 1–5% of labels (Redman, 1998; Müller and Markert, 2019; Northcutt et al., 2021b; Kreutzer et al., 2022) or even as much as 20% (Hovy et al., 2014; Abedjan et al., 2016). For organic errors, CL (Northcutt et al., 2021a) predicts errors in IMDB, Amazon Reviews, and other datasets by estimating a joint distribution between noisy and uncorrupted labels; Reiss et al. (2020) pioneers using BERT for error detection on ConLL-2003 via a classifier trained over a frozen BERT embeddings layer.

3 Methods

Motivation. Empirical evidence on image data suggests that models exhibit high loss on label errors in training data relative to the underlying features (Huang et al., 2019; Kim et al., 2021; Hong et al., 2021; Chen et al., 2021a). Hendrycks and Gimpel (2017) show that predicted probabilities of (non pre-trained) neural networks can identify out-of-distribution examples. We consider the framing that label errors are one type of out-of-distribution data. Indeed, CL (Northcutt et al., 2021a) uses normalized predicted probabilities, also from non pre-trained models, to directly identify label errors. Foundation models are highly performant; we hypothesize that a low likelihood label is likely to be an error.

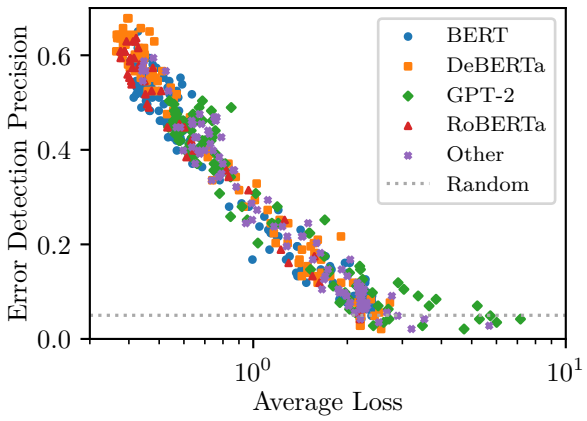

Figure 2: Loss exhibits a strong log-linear relationship with error detection precision at a fixed threshold, across a broad range of models and hyperparameters (r^2: 0.94; TweetNLP-5, §7).

Foundation models. The success of Reiss et al. (2020)'s approach in using frozen BERT embeddings motivates directly applying the foundation model paradigm: we use a large language model that was first pre-trained on a task-agnostic dataset, then fine-tune the model for a given task.

We address classification tasks: given a model's score $f_{i,c}$ for each item i and class c, its predicted probability is the softmax-normalized score $p(c \mid x_i)$. Because each item belongs to exactly one class, the contribution of item i to the loss is the negative log probability of the score for the assigned class y_i:

$$L_i = \sum_i -\log\, p(y_i \mid x_i).$$

We fine-tune such a model for the training split of each data set. To identify label errors on a validation or test set, we hypothesize items from the dataset as a label error in order of the item's loss on that out-of-distribution set.

We propose two main methods. Foundation Model Loss (FML) uses a single foundation model, fine-tuned on the corresponding task (e.g., sentiment classification, POS tagging), to hypothesize items in order of the model-predicted loss. We augment FML using task-adaptive pre-training (TAPT; Gururangan et al., 2020), which is further pre-training on in-domain data, using only text on the pre-training objective without using any labels for fine-tuning on the cross-entropy objective.

Foundation Model Ensembling (FME) combines multiple foundation models on the same

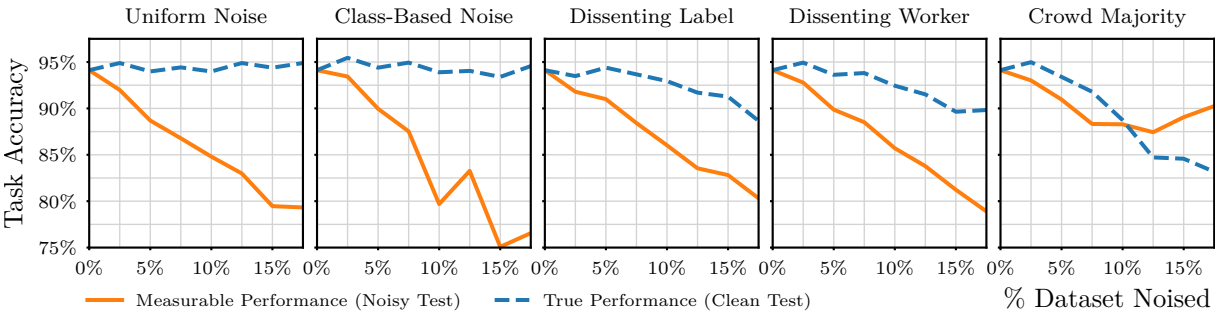

Figure 3: Assessing model robustness against a range of noising methods on TweetNLP, with methods ordered by hypothesized realism. Solid orange lines report task performance on noisy test data, reflecting observations in practice; dashed blue lines report task performance on underlying clean test data, reflecting models' actual performance. Models may be robust to uniform and class-dependent noise, where the true performance remains high even with increasing levels of noise. However, they are not necessarily robust to human-originated noise, where the true test performance decreases with increasing noise.

task. We hypothesize that ensembling may be disproportionately effective at detecting label errors, as training noise induces models to learn random spurious correlations (Watson et al., 2022). Rather than using a validation set to choose the single model with the lowest loss on the task, FME uses the top three models trained in a hyperparameter sweep, and differing in both hyperparameters and random initialization, as fully described in Appendix D. FME creates a synthetic probability distribution over the task outputs by averaging the probabilities predicted using each individual model. FME then hypothesizes items in order of loss over the synthetic distribution.

4 Generating Realistic Label Noise

To better evaluate label noise detection performance, we prepare a set of benchmark datasets populated with controllable, highly realistic, human-originated label noise.

Sources of human error. We observe that datasets often undergo multiple annotation passes: crowdsourced labels typically aggregate several annotators' inputs (Hovy et al., 2014; Wei et al., 2022), and subsets of data may receive more extensive validation (Bowman et al., 2015), gold labels by trained experts (Plank et al., 2014), or correction passes (Reiss et al., 2020). We hypothesize that differences between such annotations may be usefully repurposed as a source of realistic, *human-originated* label noise, as disagreements between annotators is known to reflect systematic ambiguity and human error (Plank et al., 2014; Zhang et al., 2017), and differs from the type of noise studied using existing synthetic methods.

We construct three noising methods which may be applied in many of the above scenarios. For any dataset which includes two levels of label quality, the *dissenting label* method replaces final labels with disagreeing labels at random, simulating imperfect quality control. Datasets which provide individual annotator identifiers may apply the *dissenting worker* approach: select one annotator at random, apply all of their labels which disagree with final labels, and repeat until reaching the target noise rate. This simulates gaps in annotator training, which introduce systematic idiosyncrasies. Finally the *crowd majority* method applies to any dataset in which individual annotations can be aggregated to produce a label other than the final label: the former label simulates challenging, systematic errors in the latter.

Noising and robustness. We assess the effect of these noising methods using TweetNLP (Gimpel et al., 2010), a corpus of 26,435 tokens from 1,827 American English tweets collected from Twitter used to train part-of-speech (POS) tagging. TweetNLP includes gold labels annotated by 17 experts, but later received a separate crowdsourced assessment, aggregated by majority vote (Hovy et al., 2014). We noise TweetNLP to eight levels from 0-20% separately for each method, fine-tune `DeBERTA-v3-base` (He et al., 2021) on each noising, and evaluate models on both noisy and clean test sets. Results from noisy test sets represent model performance as *measurable* in

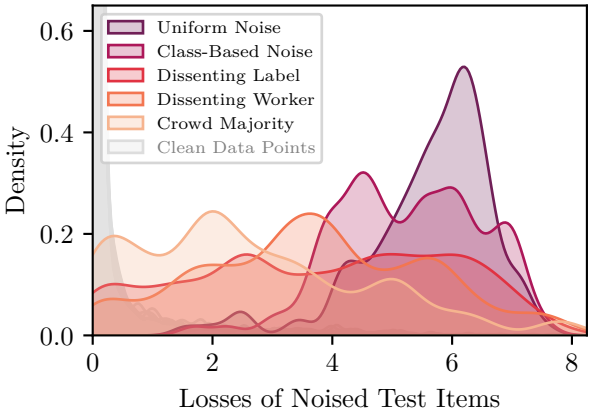

Figure 4: Distributions of losses of label errors on TweetNLP at 5% noising. Uniform and class-based noise produce high and distinctive losses; human-originated noise is widely distributed, and has greater overlap with the distribution of clean data points; §6.

practice; real datasets contend with noise in evaluation data. Clean test set results represent *true* model performance. Fig. 3 reports the results of this evaluation.

For uniform and class-dependent noise, true performance remains high even for high noise levels (per Rolnick et al., 2017). But crucially, this robustness does not extend to human-originated noise: human label errors are correlated to input text, and so contain systematic erroneous features, which models may learn in training. On more challenging noising methods, although measured performance appears to increase, true performance actually *linearly decreases* with noise. Fig. 4 explores this further via the distributions of model losses for each noising method: loss induced by human-originated noise overlaps significantly with clean items, whereas loss from uniform and class-based noising is distinctively higher.

Noise detection benchmarks. We standardize a set of benchmarks from existing datasets for use in our main experiments. TweetNLP-5 and SNLI-5 aim to simulate typical data noise conditions: we apply dissenting worker and dissenting label noising to a 5% level (see Appendix A for details). SNLI is a corpus of 570,152 sentence pairs, in which the task is to label each pair with entailment, contradiction, or semantic independence; we use the 10% subset which includes five crowdsourced annotations per item, as collected by Bowman et al. (2015) during data validation.

We construct TweetNLP-M to investigate robustness to systematic error introduced by the crowdsourcing process. We apply crowd majority noising, comparing noisy majority-vote aggregated labels by Hovy et al. (2014) to clean expert labels, which serve as a measure of true performance. Accordingly, we retain all disagreements, or 20.46% of the dataset. We also report results on Recon, a legal classification dataset of 1,279 documents in which Hong et al. (2021) found label errors to destabilize model evaluation; as above, we compare non-expert and expert annotator labels.

5 Validation on Real Label Errors

In addition to human-originated noise datasets, we evaluate error detection performance on organic errors in two benchmark datasets, following Northcutt et al. (2021a)'s protocol.

Datasets. The IMDB Large Movie Review Dataset is a collection of movie reviews for binary sentiment classification (Maas et al., 2011), and is split into train and test sets of 25,000 items each. Amazon Reviews is a collection of reviews and 5-point star ratings from Amazon customers (McAuley et al., 2015). We used the version released by Northcutt et al. (2021a), which includes the following modifications: It uses 1-star, 3-star, 5-star reviews with net positive helpful upvotes as a ternary sentiment task, resulting in a dataset of 9,996,437 reviews. For tractability we use a train split of a random sample of 2.5 million items, and a test split of 25,000 items.

Baseline protocol. Workers are presented with review text and asked to determine whether overall sentiment is positive, negative, neutral, or off-topic. Each review is independently presented to five workers. An example is considered a "Non-Error" if at least three workers agree the original label is correct. Otherwise, we consider the label to be correctly identified as an error. We further categorize label errors as "Correctable" if at least three workers agree on the same replacement label, or "Non-Agreement" if no majority exists.

New adaptations. While conducting initial experiments, we found that the Northcutt et al.

IMDB	New Protocol			
Old Protocol	C	NA	NE	Total
Correctable	105	44	24	173
Non-Agreement	75	252	225	**552**
Non-Error	3	62	520	585
Total	183	**358**	769	1310

Amazon	New Protocol			
Old Protocol	C	NA	NE	Total
Correctable	142	43	117	302
Non-Agreement	140	79	211	**430**
Non-Error	75	31	162	268
Total	357	**153**	490	1000

Table 2: Re-evaluation of baselines: The number of **C**orrectable, **N**on-**A**greement, and **N**on-**E**rror assessments produced by the CL Mechanical Turk evaluation protocol and the new protocol, on the same set of items. The new protocol substantially reduces annotator non-agreement; §5.

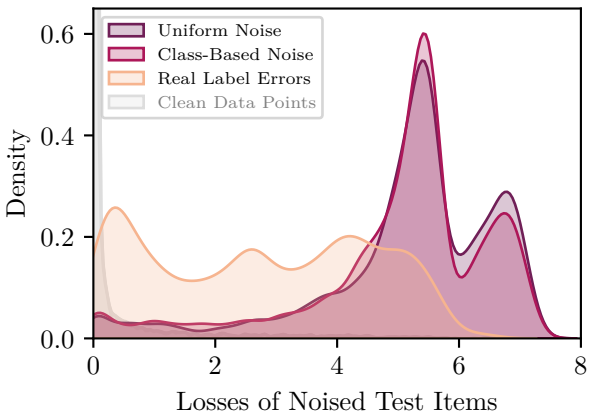

Figure 5: Distributions of losses of hypothesized label errors that MTurk workers verified for IMDB. As with Fig. 4, uniform and class-based methods do not approximate real, worker-identified errors, and losses of real label errors have greater overlap with the distribution of clean data; §6.

(2021a) MTurk protocol resulted in a significant amount of annotator fraud. Some workers spent unreasonably short amounts of time on the text, and frequently disagreed with both expert and peer annotators, reflecting increasingly common issues in crowdsourced annotations (Kennedy et al., 2020). Appendix C describes four extra conditions we added to improve the Northcutt et al. (2021a) protocol.

In order to establish an accurate baseline, we re-evaluate the label errors hypothesized by CL (Northcutt et al., 2021a). On the new protocol, Fleiss' κ inter-annotator agreement increases from 0.131 to 0.464 for IMDB, and 0.014 to 0.556 for Amazon, and Table 2 shows that Non-Agreement decreases by 35% in IMDB and 65% in Amazon. This suggests a substantial decrease in low-quality annotations.

6 Experiments

Label noise realism. Section 4 defined the human-originated noising protocol used to generate TweetNLP-5, TweetNLP-M, and SNLI-5. Section 5 specified a protocol for identifying organic label errors present in IMDB and Amazon. We assess the realism of synthetic noise methods by comparing loss distributions against models trained with organic noise (for real label errors, we refer to items verified as Correctable via MTurk). We quantify the degree to which noising induces erroneous learning by measuring the Wasserstein distances between noisy and clean loss distributions.

Overall LLM performance. We assess broad error detection capabilities by evaluating 13 commonly-used LLMs on TweetNLP-5. We measure performance against loss, model size, and GLUE score (a proxy for general model capability; Wang et al., 2018). Appendix D provides implementation details. This experiment's results inform model selection: we use `DeBERTA-v3-base` for all further experiments.[2]

Main experiment. Using our realistic noising benchmarks, and the MTurk baselines and verification protocol, we can now assess the performance of each label error detection method. We evaluate Foundation Model Loss (**FML**) and Foundation Model Ensembling (**FME**).

As a baseline, we evaluate Confident Learning (**CL**; Northcutt et al., 2021a). CL is not a standalone method; it augments existing models. Given an underlying model's predicted scores for each class and the true proportion of each class, CL forms a reweighting matrix, called the confident joint. To form a label error prediction score, CL reweights the model's scores by the confident joint. CL hypothesizes items in order of this resulting score.

CL uses FastText (Joulin et al., 2017) for IMDB and Amazon, but includes no implemen-

[2]We also use RoBERTa-BigBird for Recon in order to handle its long input passages (Hong et al., 2021).

	Area Under Precision-Recall Curve						Precision, Recall @ Error%[3]						Recall @ 2 · Error%			
	I	Am.	R	T-5	T-M	S-5	I	Am.	R	T-5	T-M	S-5	R	T-5	T-M	S-5
H&G	-	-	-	0.30	0.41	0.20	-	-	-	0.31	0.44	0.22	-	0.54	0.63	0.34
CL	0.24	0.31	0.25	0.30	0.41	0.17	0.41	0.51	0.31	0.36	0.44	0.18	0.46	0.47	0.63	0.32
FML	0.58	0.39	0.37	0.66	*0.48*	0.54	0.68	0.64	**0.46**	0.65	0.47	0.45	0.62	0.88	0.64	0.66
FME	**0.60**	**0.40**	**0.38**	*0.68*	*0.48*	0.61	**0.69**	**0.66**	0.38	0.66	*0.48*	0.46	*0.69*	0.88	0.65	*0.68*
FME+CL	0.20	0.17	0.37	*0.68*	*0.48*	**0.62**	-	-	0.38	**0.69**	*0.48*	**0.47**	*0.69*	**0.89**	**0.66**	*0.68*

Table 3: Main experiment: Evaluating label error detection methods using datasets containing highly-realistic label errors (**I**MDB, **Am**azon Reviews, **R**econ, **T**weetNLP-**5**, **T**weetNLP-**M**, **S**NLI-**5**). Foundation model-based methods significantly outperform baselines on every dataset, as shown by an overall performance metric (AUPR). In practice, estimating the number of dataset errors and checking this many items quickly catches up to 69% of errors, at the same accuracy (P,R@Err%).[3] For improved coverage, checking twice this number of items catches up to 89% of errors (R@2·Err%).

tations for POS tagging or NLI. As a result, for TweetNLP and SNLI, we apply CL to the **H&G** baseline (Hendrycks and Gimpel, 2017), a two-layer neural classifier over word vectors pre-trained on a corpus of 56 million tweets (Owoputi et al., 2013). For all datasets, we also assess applying CL to foundation models (**FME+CL**).

For each dataset, we run 25 hyperparameter sweeps which each fine-tune a model for the given task (e.g., POS tagging) using noisy data, and select the model with the best validation set task performance. We report label error detection performance (not task performance). Area Under the Precsion-Recall Curve (AUPR) provides an overall performance score (Saito and Rehmsmeier, 2015; Hendrycks and Gimpel, 2017). We also report metrics representing performance on competing data cleaning priorities: efficiency requires high precision on a small number of items, whereas coverage requires high recall on a larger number of items. Appendix E.1 describes the Truncated AUPR used for IMDB and Amazon, which are too costly to fully crowd verify.

End-to-end noising. We finally isolate the effects of noise and label error correction for validation and test splits. For each dataset, we prepare three versions of the validation and test splits, respectively: a *clean* version assumed to contain zero errors,[4] a *noisy* version, with label noise deliberately introduced, and a *corrected* version generated from noisy splits using our main error detection method (ranking errors with FME and correcting the top Err% data points). We train 40 hyperparameter sweeps, with performance cross-evaluated on all prepared data splits.

We report three different metrics. We report each model's accuracy on the clean test split as the *true* accuracy. Following the norms of Fig. 3, we report the *measurable* accuracy as the accuracy of the model selected using performance on the noisy or corrected validation split on the corresponding test split. Finally, we report the *rank* of the model as the rank of the model's performance on clean test data. The best performing model among all sweeps has rank 1, and the worst has rank 40. This metric emphasizes that different validation sets select different models.

We perform this exercise using IMDB and Amazon noised to 5% (I-5, A-5), and TweetNLP-5 and TweetNLP-M.

7 Results

Label noise realism. Human-originated noise appears to closely approximate real label noise. Figs. 4 and 5 show that the losses of both real and human-originated label errors are lower and more widely-distributed than existing noising methods. Their Wasserstein distances to the distribution of clean data are significantly lower than existing noising methods, suggesting comparable erroneous learning (Appendix B).

Overall LLM performance. We discover a strong log-linear relationship between error

[3]Precision and recall are equal when evaluating a number of items equal to the total error count.

[4]For TweetNLP, we justify our assumption in §4: expert labels by Hovy et al. (2014) are considered noise free compared to crowd labels. For IMDB and Amazon, we follow Northcutt et al. (2021a), which adds several percentage points more noise than naturally occurs.

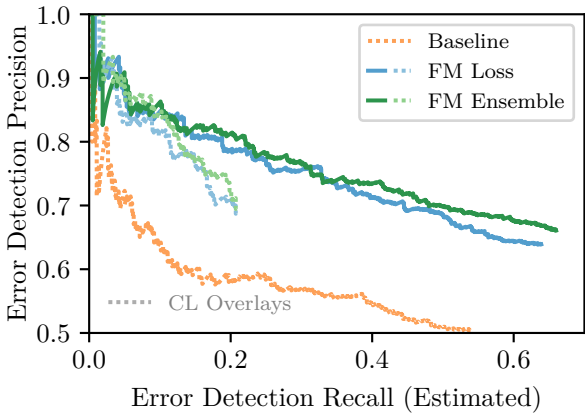

Figure 6: Precision-recall curves for label error detection on Amazon by method. FML+CL and FME+CL produce fewer items and do not extend to a recall past 0.21. Applying CL to FM changes little compared to using FM alone.

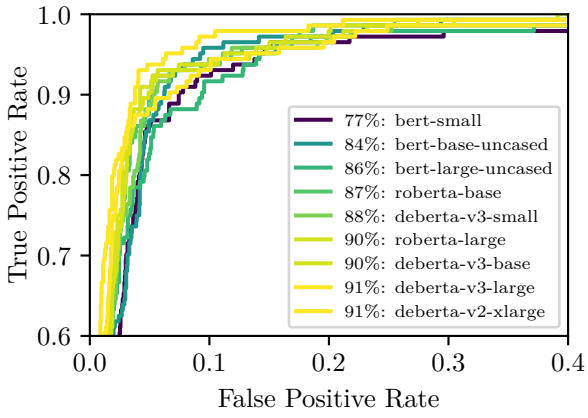

Figure 7: ROC curves for error detection performance on TweetNLP-5: LLM loss is highly effective for detecting label errors, and performance is highly correlated with general language understanding (GLUE, r^2: 0.79).

detection performance and loss, which holds across many model families and configurations (r^2: 0.94, Fig. 2). We also find relationships between error detection performance and general model capability, in terms of GLUE score (r^2: 0.79) and model size (Fig. 10). Fig. 7 illustrates key findings using models' receiver operating characteristic (ROC) curves. Ensembling confers signficantly more gains in error detection performance higher than gains on underlying task performance, across a broad range of models and hyperparameters; Appendix E.3 explores ensembling in greater detail.

Main experiment. Table 3 shows that Foundation Model Ensembling significantly improves AUPR from the CL and H&G baselines on all datasets, with an absolute difference of 0.36 on IMDB, 0.09 on Amazon, and a difference of 0.07–0.44 on synthetic data.

Fig. 1 shows that applying CL to FME has minimal effect on performance at every level of recall; most numbers are identical across the FME and FME+CL rows of Table 3. In fact, CL does not necessarily improve upon the H&G baseline across datasets, with CL performance sometimes dipping below H&G by 0.01–0.03.

While loss naturally ranks all data points, CL only hypothesizes a fixed number of potential errors: Appendix E.2 shows the raw counts of items at fixed thresholds, per the original CL study. At the CL threshold, we outperform CL by an absolute 15–28%. At the CL+FME threshold, predicted items are almost exactly the same, with Jaccard similarities of 0.59–0.99. By contrast, ensembling improves performance over FML by a greater amount on almost every measure, and introduces no such constraint.

End-to-end noising. Cleaning validation data selects better models. Noise in validation splits reduces performance by encouraging the selection of models with lower true performance. Noise in test splits significantly reduces measureable (noisy test) performance, as expressed by the difference between measureable and true performance. In general, correcting label errors improves task performance: even when the reported task performance worsens, the reported performance is closer to the true performance of the model, measured using clean training and validation data.

8 Discussion

Rapid data "health check". Sorting evaluation data by each item's loss is an easy way to quickly highlight label errors. Using this simple technique with a foundation model appears to generally identify over half of all label errors through human re-evaluation of a single-digit percentage of all data (Table 3). We expect this technique to work across deep learning domains, due to its simplicity and the extensive use of training loss in LNL research (Song et al., 2022). Given estimates for typical rates of label errors and the gain observed in the end-to-end experiment, our technique may enable a 1–2% increase in reportable test accuracy

Eval.	Test Perf.	I-5	A-5	T-5	T-M
Noisy	Measurable	90.1	88.3	89.3	89.3
	True	94.2	91.0	92.8	82.0
	Rank	10	1	3	10
Corr.	Measurable	95.1	90.7	92.9	88.5
	True	95.1	90.8	93.0	82.0
	Rank	4	5	2	8
Clean	True	95.8	91.0	93.8	82.1

Table 4: End-to-end effects of label noise on task performance, as evaluated on noisy, corrected, and clean validation and test data splits. True accuracy is measured on clean test sets, and measurable accuracy on noisy or corrected test sets. Rank is a relative measure of true accuracy; lower numerical ranks have higher accuracy. Corrections which improve or reduce performance metrics are highlighted in green or red, respectively. Metrics are evaluated on models trained on noisy data.

across many datasets, in addition to the gains from improving model selection.

Pre-training and robustness. We demonstrate that despite established findings on artificial noising (Hendrycks et al., 2018), pre-training confers limited robustness to realistic human noise. The majority of label errors are systematic in nature (Snow et al., 2008; Plank et al., 2014; Samuel et al., 2022), and crowdsourced labels form, to an extent, a different distribution from reality, as approximated by expert labels (Hendrycks et al., 2020). When trained on crowdsourced or other data containing systematic errors, FMs quickly drift towards this incorrect distribution.

Applying AI to data-centric AI. Data-centric AI aims to improve AI through labeling, curating, and augmenting the underlying data. We find that AI itself can be applied towards improving data quality, as part of a human-in-the-loop (HITL) iteration, which contributes an additional positive feedback loop between data quality and AI performance.

New challenges in LNL. Standard noising methods are unrealistic and no longer challenging for state-of-the-art language models (Algan and Ulusoy, 2020); recent LNL analyses study conditions where up to 80% of labels are noised (Song et al., 2022). Our findings reinforce the need to reassess LNL methods in the context of more realistic noise (Zhu et al., 2022).

Our human-originated noising method produces realistic label errors, and can be applied to any crowdsourced dataset which includes raw annotation data. As such datasets emerge across deep learning domains (Wei et al., 2022), we hope this method may inspire challenging and realistic new LNL performance benchmarks. Our method also enables detailed exploration of the properties of human noise, which may support work on open LNL problems such as improving feature-based noising techniques, and estimating dataset noise (Bäuerle et al., 2022; Northcutt et al., 2021b).

End-to-end noising. The study of model performance on noise in validation and test data is essential: noise in other splits can affect reported model performance as much as noise in training data. Clean and noisy performance on evaluation data provide useful insight into models' overall performance.

9 Conclusions and Future Work

Pre-trained models effectively identify label errors on real NLP datasets, definitively outperforming existing methods on the same benchmarks by an absolute 9–36% in AUPR.

Human-originated noising techniques may present a solution to the clear limitations of current LNL noising schemes: they are highly realistic and yet controllable for experimental purposes. We invite further exploration of this family of label noising techniques. We believe human-originated noising enables future advancements across multiple areas of LNL, supporting new tasks and metrics in areas such as the cost of human reannotation, estimation of dataset error, and mitigation of bias.

Finally, we advocate for LNL to move towards an end-to-end approach of *evaluating with label noise*, which takes into account noise within validation and test splits, and more accurately models the conditions of data in practice.

Limitations

Partial metrics. Determining the true recall of a label error detection method on a real datasets is generally infeasible due to its high cost; this requires a complete re-evaluation so as to identify every label error within the dataset. While some datasets exist in which this has been undertaken, such as Hovy et al.

(2014) for TweetNLP, for most datasets containing organic label errors, we can only assess precision directly.

To mitigate this, we can estimate recall by estimating total dataset error counts using sampling techniques. As a result of this limitation, we prefer AUPR over AUROC (Area Under the Receiving Operating Curve) as our overall assessment metric: estimates of AUPR are scaled by a fixed ratio, and therefore comparable between models on the same dataset, whereas AUROC is nonlinear with respect to the estimate.

Requires multiple annotations per label. Human-originated noising methods are only applicable to datasets which include at least two human annotations per label. While it is becoming increasingly common to release individual-level annotator data, this is not an ubiquitous practice.

Cleaning benchmark data. In our analysis of model performance gains derived from applying our methods to cleaning evaluation data, we find that cleaning validation splits enables the selection of models with better test performance. Such a method may be useful in a large number of applications.

However, we caution against using this method to clean data intended for use in comparing performance across model families and variants: the cleaning process may bias any such benchmarks toward the models most similar to the model used to clean the data. While our method improves the performance of a given model on a task, and correcting label errors always improves the validity of test data, these improvements is unlikely to improve the performance of all models by the same amount.

This limitation is shared with other existing model-based scoring methods such as BERTScore (Zhang* et al., 2020).

Acknowledgements

We would like to thank Google.org for credits for use of the Google Cloud Platform. We are also grateful to our anonymous reviewers, members of the Stanford NLP Group, and Bryan H. Chong for their constructive feedback, as well as the many researchers who made data publicly available to enable our present work.

References

Ziawasch Abedjan, Xu Chu, Dong Deng, Raul Castro Fernandez, Ihab F Ilyas, Mourad Ouzzani, Paolo Papotti, Michael Stonebraker, and Nan Tang. 2016. Detecting data errors: Where are we and what needs to be done? *Proceedings of the VLDB Endowment*, 9(12):993–1004.

Jon Agley, Yunyu Xiao, Rachael Nolan, and Lilian Golzarri-Arroyo. 2021. Quality control questions on Amazon's Mechanical Turk (MTurk): A randomized trial of impact on the USAUDIT, PHQ-9, and GAD-7. *Behavior research methods*, pages 1–13.

Görkem Algan and Ilkay Ulusoy. 2020. Label noise types and their effects on deep learning. *arXiv:2003.10471*.

Ehsan Amid, Manfred KK Warmuth, Rohan Anil, and Tomer Koren. 2019. Robust bi-tempered logistic loss based on Bregman divergences. *Advances in Neural Information Processing Systems*, 32.

Samaneh Azadi, Jiashi Feng, Stefanie Jegelka, and Trevor Darrell. 2015. Auxiliary image regularization for deep CNNs with noisy labels. *arXiv preprint arXiv:1511.07069*.

Noga Bar, Tomer Koren, and Raja Giryes. 2021. Multiplicative reweighting for robust neural network optimization. *arXiv preprint arXiv:2102.12192*.

Alex Bäuerle, Ángel Alexander Cabrera, Fred Hohman, Megan Maher, David Koski, Xavier Suau, Titus Barik, and Dominik Moritz. 2022. Symphony: Composing interactive interfaces for machine learning. In *CHI Conference on Human Factors in Computing Systems*, pages 1–14.

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. 2021. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*.

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*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Jiefeng Chen, Frederick Liu, Besim Avci, Xi Wu, Yingyu Liang, and Somesh Jha. 2021a. Detecting errors and estimating accuracy on unlabeled data with self-training ensembles. *Advances in Neural Information Processing Systems*, 34.

Pengfei Chen, Junjie Ye, Guangyong Chen, Jingwei Zhao, and Pheng-Ann Heng. 2021b. Beyond

class-conditional assumption: A primary attempt to combat instance-dependent label noise. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 11442–11450.

Glenn Dawson and Robi Polikar. 2021. Rethinking noisy label models: Labeler-dependent noise with adversarial awareness. *arXiv preprint arXiv:2105.14083*.

DCAI Workshop. 2021. NeurIPS data-centric AI workshop. *NeurIPS 2021 Data-Centric AI Workshop*.

Sarah Jane Delany, Nicola Segata, and Brian Mac Namee. 2012. Profiling instances in noise reduction. *Knowledge-Based Systems*, 31:28–40.

Sean A Dennis, Brian M Goodson, and Christopher A Pearson. 2020. Online worker fraud and evolving threats to the integrity of MTurk data: A discussion of virtual private servers and the limitations of IP-based screening procedures. *Behavioral Research in Accounting*, 32(1):119–134.

Shrey Desai and Greg Durrett. 2020. Calibration of pre-trained transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 295–302, Online. Association for Computational Linguistics.

Benoit Frenay and Michel Verleysen. 2014. Classification in the presence of label noise: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 25(5):845–869.

Dragan Gamberger, Nada Lavrac, and Saso Dzeroski. 2000. Noise detection and elimination in data preprocessing: experiments in medical domains. *Applied artificial intelligence*, 14(2):205–223.

Kevin Gimpel, Nathan Schneider, Brendan O'Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and Noah A Smith. 2010. Part-of-speech tagging for Twitter: Annotation, features, and experiments. Technical report, Carnegie-Mellon Univ Pittsburgh Pa School of Computer Science.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. DeBERTaV3: Improving DeBERTa using Electra-style pre-training with gradient-disentangled embedding sharing. *arXiv:2111.09543*.

Dan Hendrycks and Kevin Gimpel. 2017. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *Proceedings of International Conference on Learning Representations*.

Dan Hendrycks, Xiaoyuan Liu, Eric Wallace, Adam Dziedzic, Rishabh Krishnan, and Dawn Song. 2020. Pretrained transformers improve out-of-distribution robustness. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2744–2751, Online. Association for Computational Linguistics.

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.

Jenny Hong, Derek Chong, and Christopher Manning. 2021. Learning from limited labels for long legal dialogue. In *Proceedings of the Natural Legal Language Processing Workshop 2021*, pages 190–204, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Dirk Hovy, Barbara Plank, and Anders Søgaard. 2014. Experiments with crowdsourced re-annotation of a POS tagging data set. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 377–382, Baltimore, Maryland. Association for Computational Linguistics.

Jinchi Huang, Lie Qu, Rongfei Jia, and Binqiang Zhao. 2019. O2u-net: A simple noisy label detection approach for deep neural networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3326–3334.

Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. 2018. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *International Conference on Machine Learning*, pages 2304–2313. PMLR.

Yiding Jiang, Vaishnavh Nagarajan, Christina Baek, and J Zico Kolter. 2022. Assessing generalization of SGD via disagreement. In *International Conference on Machine Learning*.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas 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: Volume 2, Short Papers*, pages 427–431. Association for Computational Linguistics.

Armand Joulin, Laurens van der Maaten, Allan Jabri, and Nicolas Vasilache. 2016. Learning visual features from large weakly supervised data. In *European Conference on Computer Vision*, pages 67–84. Springer.

Shyamgopal Karthik, Jérome Revaud, and Boris Chidlovskii. 2021. Learning from long-tailed data with noisy labels. *arXiv:2108.11096*.

Ryan Kennedy, Scott Clifford, Tyler Burleigh, Philip D Waggoner, Ryan Jewell, and Nicholas JG Winter. 2020. The shape of and solutions to the MTurk quality crisis. *Political Science Research and Methods*, 8(4):614–629.

Taehyeon Kim, Jongwoo Ko, JinHwan Choi, Se-Young Yun, et al. 2021. FINE samples for learning with noisy labels. *Advances in Neural Information Processing Systems*, 34.

Julia Kreutzer, Isaac Caswell, Lisa Wang, Ahsan Wahab, Daan van Esch, Nasanbayar Ulzii-Orshikh, Allahsera Tapo, Nishant Subramani, Artem Sokolov, Claytone Sikasote, et al. 2022. Quality at a glance: An audit of web-crawled multilingual datasets. *Transactions of the Association for Computational Linguistics*, 10:50–72.

Abhishek Kumar and Ehsan Amid. 2021. Constrained instance and class reweighting for robust learning under label noise. *arXiv*.

Yang Liu and Hongyi Guo. 2020. Peer loss functions: Learning from noisy labels without knowing noise rates. In *International Conference on Machine Learning*, pages 6226–6236. PMLR.

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

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

Hassan H Malik and Vikas S Bhardwaj. 2011. Automatic training data cleaning for text classification. In *2011 IEEE 11th international conference on data mining workshops*, pages 442–449. IEEE.

Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton van den Hengel. 2015. Image-based recommendations on styles and substitutes. In *SIGIR 2015*, page 43–52, New York, NY, USA. Association for Computing Machinery.

Alexandra M Mellis and Warren K Bickel. 2020. Mechanical Turk data collection in addiction research: Utility, concerns and best practices. *Addiction*, 115(10):1960–1968.

Aaron Moss and Leib Litman. 2018. After the bot scare: Understanding what's been happening with data collection on MTurk and how to stop it. *CloudResearch*.

Nicolas M Müller and Karla Markert. 2019. Identifying mislabeled instances in classification datasets. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

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

Curtis G Northcutt, Anish Athalye, and Jonas Mueller. 2021b. Pervasive label errors in test sets destabilize machine learning benchmarks. *arXiv:2103.14749*.

Olutobi Owoputi, Brendan O'Connor, Chris Dyer, Kevin Gimpel, Nathan Schneider, and Noah A. Smith. 2013. Improved part-of-speech tagging for online conversational text with word clusters. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 380–390, Atlanta, Georgia. Association for Computational Linguistics.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Barbara Plank, Dirk Hovy, and Anders Søgaard. 2014. Linguistically debatable or just plain wrong? In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 507–511.

Thomas C Redman. 1998. The impact of poor data quality on the typical enterprise. *Communications of the ACM*, 41(2):79–82.

Scott Reed, Honglak Lee, Dragomir Anguelov, Christian Szegedy, Dumitru Erhan, and Andrew Rabinovich. 2014. Training deep neural networks on noisy labels with bootstrapping. *arXiv preprint arXiv:1412.6596*.

Frederick Reiss, Hong Xu, Bryan Cutler, Karthik Muthuraman, and Zachary Eichenberger. 2020. Identifying incorrect labels in the CoNLL-2003 corpus. In *Proceedings of the 24th conference on computational natural language learning*, pages 215–226.

David Rolnick, Andreas Veit, Serge Belongie, and Nir Shavit. 2017. Deep learning is robust to massive label noise. *arXiv:1705.10694*.

Takaya Saito and Marc Rehmsmeier. 2015. The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PloS one*, 10(3):e0118432.

Jim Samuel, Gavin Rozzi, and Ratnakar Palle. 2022. The dark side of sentiment analysis: An exploratory review using lexicons, dictionaries, and a statistical monkey and chimp. *Dictionaries, and a Statistical Monkey and Chimp.(January 6, 2022)*.

Antonios Saravanos, Stavros Zervoudakis, Dongnanzi Zheng, Neil Stott, Bohdan Hawryluk, and Donatella Delfino. 2021. The hidden cost of using Amazon Mechanical Turk for research. In *International Conference on Human-Computer Interaction*, pages 147–164. Springer.

Borut Sluban, Dragan Gamberger, and Nada Lavrač. 2014. Ensemble-based noise detection: noise ranking and visual performance evaluation. *Data mining and knowledge discovery*, 28(2):265–303.

Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Ng. 2008. Cheap and fast – but is it good? evaluating non-expert annotations for natural language tasks. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 254–263, Honolulu, Hawaii. Association for Computational Linguistics.

Hwanjun Song, Minseok Kim, Dongmin Park, Yooju Shin, and Jae-Gil Lee. 2022. Learning from noisy labels with deep neural networks: A survey. *IEEE Transactions on Neural Networks and Learning Systems*.

Jaree Thongkam, Guandong Xu, Yanchun Zhang, and Fuchun Huang. 2008. Support vector machine for outlier detection in breast cancer survivability prediction. In *Asia-Pacific Web Conference*, pages 99–109. Springer.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Matthew Watson, Bashar Awwad Shiekh Hasan, and Noura Al Moubayed. 2022. Agree to disagree: When deep learning models with identical architectures produce distinct explanations. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 875–884.

Jiaheng Wei, Zhaowei Zhu, Hao Cheng, Tongliang Liu, Gang Niu, and Yang Liu. 2022. Learning with noisy labels revisited: A study using real-world human annotations. In *International Conference on Learning Representations*.

Virginia Wheway. 2000. Using boosting to detect noisy data. In *Pacific Rim International Conference on Artificial Intelligence*, pages 123–130. Springer.

Liang Xu, Jiacheng Liu, Xiang Pan, Xiaojing Lu, and Xiaofeng Hou. 2021. DataCLUE: A benchmark suite for data-centric NLP. *arXiv:2111.08647*.

Jing Zhang, Victor S Sheng, Qianmu Li, Jian Wu, and Xindong Wu. 2017. Consensus algorithms for biased labeling in crowdsourcing. *Information Sciences*, 382:254–273.

Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating text generation with BERT. In *International Conference on Learning Representations*.

Wenzheng Zhang and Karl Stratos. 2021. Understanding hard negatives in noise contrastive estimation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1090–1101, Online. Association for Computational Linguistics.

Wenxuan Zhou and Muhao Chen. 2021. Learning from noisy labels for entity-centric information extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5381–5392, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Dawei Zhu, Michael A Hedderich, Fangzhou Zhai, David Ifeoluwa Adelani, and Dietrich Klakow. 2022. Is BERT robust to label noise? A study on learning with noisy labels in text classification. *arXiv:2204.09371*.

A Noising Benchmarks

This section specifies how noising protocols were applied to create each fixed crowdsourced dataset. Crowd labels for each dataset are available to download from the respective GitHub projects.

A.1 TweetNLP-5

TweetNLP-5(%) is a fixed noising of TweetNLP to a 5% noise level in each split. Of the label errors, 80% (i.e. 4% of each split) are assigned using the *dissenting worker* method. The remaining 20% (i.e. 1% of each split) are assigned

using the *dissenting label* method. Fig. 4 shows that both methods provide similar distributions of label errors. Although the dissenting worker method more realistically captures individual worker idiosyncrasies, the dissenting label method is actually slightly lower loss during training (i.e. harder for a model to distinguish from correct labels).

A.2 TweetNLP-M

TweetNLP-M(ajority) directly uses the majority class labels collected by Hovy et al. (2014) on the Crowdflower platform, which have a 79.54% agreement with the high-quality expert gold labels collected by Gimpel et al. (2010). Per the Hovy et al. (2014) protocol, in the rare case of ties, the tie is broken in favor of the label that matches the gold label, if applicable. Otherwise, a label is selected at random. The "-M" suffix distinguishes the Hovy et al. (2014) labels from the gold labels.

A.3 SNLI-5

The Stanford Natural Language Inference dataset (SNLI) annotations do not include a worker identifier, meaning each item is attached to five crowdsourced labels, but there is no indication of which labels came from the same annotator across the dataset. As a result, we cannot apply the dissenting worker noising method.

SNLI-5 has exactly 5% of its data noised in each split. Of the label errors, 80% (i.e. 4% of each split) are assigned using a method that represents systematic errors, to simulate of dissenting worker method: We use the minority label when there is a 3-2 split between the five labels. The remaining 20% (i.e. 1% of each split) are assigned using the dissenting label method, as in TweetNLP-5.

B Loss Distributions

Section 4 examines dataset noisings primarily in terms of loss distributions on noised labels. To provide additional context, Fig. 8 provides an equivalent view for SNLI, and Fig. 9 shows combined distributions of both clean and noisy data points on TweetNLP.

Table 5 reports the Wasserstein distances (or earth mover's distances) measured between the loss distributions of noisy and clean data points for models trained on TweetNLP and IMDB, as

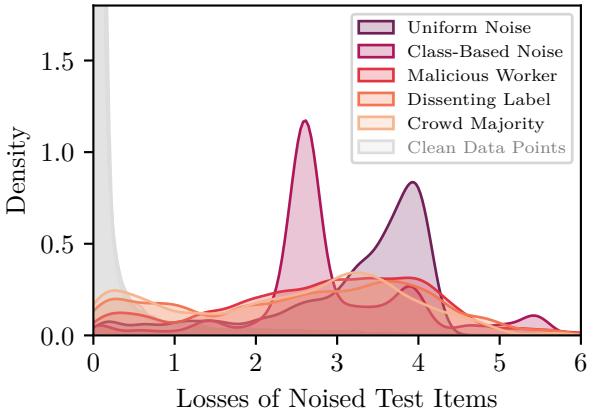

Figure 8: Distributions of losses of label errors on SNLI at 5% noising, which demonstrates similar performance characteristics to TweetNLP, as shown in Fig. 4.

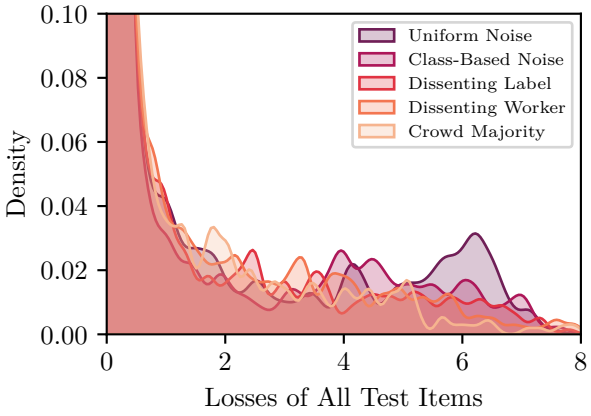

Figure 9: Combined distributions of losses of both noisy and clean data points, for TweetNLP with 5% noising.

described in Section 7. Human-originated label noise more closely resembles both clean data points and real label noise as its hypothesized realism increases.

C Mechanical Turk Protocol

C.1 Change Specifications

We use Amazon Mechanical Turk to validate real label errors from IMDB (Maas et al., 2011) and Amazon Reviews (McAuley et al., 2015). We begin with the Northcutt et al. (2021a) protocol, and add four additional conditions, so as to mitigate annotator fraud.

First, we pre-qualify workers by requiring them to correctly answer a qualification test of four unambiguous questions (Hovy et al., 2014; Agley et al., 2021).

Second, after the initial qualification, we con-

Noising Method	TweetNLP	IMDB
Uniform Noise	5.62	5.04
Class-Based Noise	5.23	4.91
Dissenting Label	4.02	-
Dissenting Worker	3.44	-
Crowd Majority	**2.33**	-
Real Label Errors	-	**2.67**

Table 5: Wasserstein distances between loss distributions of noisy and clean data points: Human-originated noising exhibits comparable levels of erroneous learning to organic label errors.

	IMDB	Amazon
Original Protocol	0.1314	0.0141
New Protocol	**0.4643**	**0.5561**

Table 6: A comparison of inter-annotator agreement between the original and new MTurk protocol results using Fleiss' κ. A score of 1.0 represents perfect agreement between workers, and 0.0 represents guessing at random. Annotations from the original protocol are substantially closer to random chance.

tinue to monitor worker quality by introducing sentinel questions with known answers into the workers' regular tasks. We periodically remove workers who fail the tasks.

Third, we set filter criteria to limit workers to the following Anglosphere countries: United States, Canada, United Kingdom, Ireland, Australia, and New Zealand (Moss and Litman, 2018), to improve the chances of finding annotators with sufficient cultural context to correctly interpret review text.[5] Our filter criteria include the standard recommendations of requiring a $\geq$ 99% positive task approval rate with $\geq$ 500 tasks approved.

Finally, we set a baseline target rate of US$10 per hour, calculated using word counts and average reading speed (primarily for ethical reasons; the effect of compensation and annotation quality is an area of active research; Saravanos et al., 2021).

The new protocol's labels are produced using a final set of approximately 70 workers. Workers averaged at least 12 seconds on each task; half the time needed to read prompts at an average reading speed. The average time spent by a worker in the Northcutt et al. (2021a) protocol was 5 seconds.[6]

C.2 Protocol Validation

We hypothesize that the Non-Agreements in the original protocol represent not only ambiguous data points, but also noise in the original protocol resulting from low quality work. Tables 2 and 6 show that the new protocol improves the level of agreement between workers. As such, we confirm that the increased agreement between workers in the new protocol results from higher quality labels.

Following the Northcutt et al. (2021a) protocol for expert review, we additionally select a total of 50 items from each of IMDB and Amazon for expert review. The experts are blinded to both the original labels and MTurk results and asked to label each item from scratch. They then reconciled results and came to a consensus for each item. The results are compared at the aggregate level of "Correctable," "Non-Agreement," and "Non-Error," as opposed to the individual sentiment level (Positive, Negative, Neutral, or Off-Topic). The expert agreement with one another was 79%, so in 21% of the items, the expert label was considered to be Non-Agreement and matched the MTurk workers only if the workers also produced Non-Agreement. Table 7 provides the result of this assessment.

For the original protocol, 52% of the items agreed with expert annotators, 31% of the items were incorrectly labeled as Non-Agreement, 12% of the items were incorrectly labeled as Correctables, and 5% of the items were incorrectly labeled as Non-Errors. 8% of items were disagreements between experts and crowd workers where neither side had a Non-Agreement. In other words, 8% of all items were disagreements between Correctable and Non-Error.

For the new protocol, 72% of the items agreed with expert annotators, 4% of the items were incorrectly labeled as Non-Agreement, 7% of the items were incorrectly labeled as Correctable, and 17% were incorrectly labeled as

[5] Despite these precautions, we recognize that every precaution is subject to fraud, e.g., location is subject to VPN and bot attacks. (Dennis et al., 2020; Mellis and Bickel, 2020; Kennedy et al., 2020)

[6] The reported time is an *upper* bound on the average time a worker spends on a task.

	IMDB	Amazon	Total
Original Correct	33	19	52
New Correct	**41**	**31**	72
Both Correct	28	14	42

Table 7: A comparison of original and new MTurk protocol results against 100 expert-labeled data points.

Non-Errors. 5% of items were disagreements between experts and crowd workers where neither side had a Non-Agreement.

D Overall LLM Performance Experiments

Due to the high costs associated with expert and crowdsourced validation, we use TweetNLP-5 as a development dataset for model selection.

We selected the following models for exploration: XLNet (`base`, `large`), RoBERTa (`base`, `large`), BERT (`small`, `base`, `large`), DeBERTa (V3: `xsmall`, `small`, `base`, `large`, and V2: `xlarge`, `xxlarge`), GPT (assorted). We performed 25 hyperparameter sweeps with each model, selecting the top three runs for further analysis. In order to avoid model family-level bias in the choice of hyperparameters, we set a broad shared range for three hyperparameters: learning rate varying from 10^{-6} to 10^{-3}, the number of epochs from 2 to 8, and the batch size between 8, 16, 64, and 128. Training time and the final hyperparameters varied based on the model.

We ultimately selected `DeBERTA-v3-base` as a compromise between performance and training speed. We used Google Cloud Platform for training infrastructure. Experiments were run using NVIDIA A100 GPUs, and runtime per training run was approximately 20 minutes for IMDB, Recon, and SNLI, 3 minutes for TweetNLP, and 4 hours for Amazon, when configured with a 2.5 million data point training split.

E Main Experiment

E.1 Metrics

We calculate the Area Under the Precsion-Recall Curve (AUPR) using the trapezoidal rule, given individual measurements of precision and recall at every possible threshold.

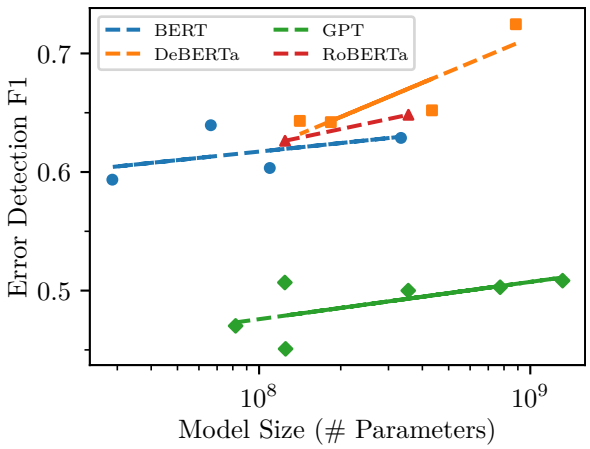

Figure 10: Label noise detection performance by model size and family, evaluated on TweetNLP-5. GPT-based models exhibit similar scaling trends, despite intrinsic disadvantages on classification tasks (due to pure autoregressive pre-training).

We report the Truncated AUPR on IMDB and Amazon. Because IMDB and Amazon are too expensive to fully crowd verify, we cannot calculate precision and recall at the 25,000th item for each method, for each dataset, as it would require every data point to be relabeled on MTurk. Instead, we use the CL framework of predicting a fixed number of items. For example, for IMDB, CL hypothesizes 1,310 out of the 25,000 items to be label errors. We can calculate the precision and recall for every threshold, up to the number hypothesized by Confident Learning. We can calculate the precision and recall of the 1st, 2nd, 3rd, . . . , and 1,310th items.

We know the exact recall for all synthetic datasets. For IMDB and Amazon, we use the estimate that 5% of the data is erroneous, which is consistent with common understanding of the prevalence of label errors (Redman, 1998; Müller and Markert, 2019; Northcutt et al., 2021b; Kreutzer et al., 2022).

All results reported on synthetic datasets reflect the average of individual scores from the three top-performing models from 25 hyperparameter sweeps. However, for cost-efficiency, results which require crowdsourced evaluation (such as IMDB and Amazon) are based on one run selected at random from a top three.

E.2 Confident Learning

Northcutt et al. (2021b) reports results using raw counts, not the accuracy, precision, recall,

Dataset	Num. Errors Hypothesized	Correctable			Non-Agreement			Non-Error		
		CL	FML	FME	CL	FML	FME	CL	FML	FME
IMDB	1310	183	323	**328**	358	573	581	769	414	401
Amazon	1000	357	508	**517**	148	131	143	495	361	340
TweetNLP-M	250	121	158	**165**	-	-	-	129	92	85

Table 8: The number of each type of error accurately identified for each dataset by each noise detection method, keeping the number of errors hypothesized fixed for ease of comparison. (TweetNLP is expert reviewed and by construction does not have any Non-Agreement types.)

Dataset	Num. Errors Hypothesized	Correctable		Non-Agreement		Non-Error		Jaccard Similarity
		FME	FME+CL	FME	FME+CL	FME	FME+CL	
IMDB	316	168	168	108	108	40	40	0.99
Amazon	381	**226**	204	65	56	90	121	0.60
TweetNLP-M	129	93	**98**	-	-	36	31	0.59

Table 9: Examining the performance of overlaying Confident Learning on FME, comparing the number of errors hypothesized by FME+CL. We also report the Jaccard similarity between the two models.

or any other metric. For ease of comparability, Table 8 reports the number of correctable, non-agreement, and non-error items identified by each method on each dataset. CL hypothesizes a fixed number of items, which is reported in the last column, and we assess a matching number of items from each method.

When hypothesizing a fixed number of items, the foundation model approaches far outperform CL baselines. On IMDB, FME correctly identifies 909 label errors, a 28% absolute improvement in accuracy. On Amazon, the FME approach correctly identifies 660 label errors, compared to the 505 identified by CL, a 15.5% absolute improvement.

Applying CL to FME results in a different model that hypothesizes a different number of items (fewer, in all cases). Table 9 shows the raw counts of correctable, non-agreement, and non-error items when each of our models hypothesizes items at this reduced threshold.

Overlaying CL on foundation model loss appears to have little marginal utility. Table 9 also shows a high Jaccard similarity across all datasets, suggesting that applying CL on top of an FM changes little about the items hypothesized. On many datasets, FME and FME+CL perform almost identically in the number of items correctly hypothesized, slightly harming performance on Amazon Reviews, and slightly improving it on TweetNLP-5 (Table 10). FME+CL decreases the total number of hypothesized items compared to FME because of the threshold set by CL. We compare the FME and FME+CL approaches at the reduced number of hypothesized items in order to assess the impact of CL in the presence of pre-training.

Not only is aggregate performance nearly identical, we see in Figs. 1 and 6 that FME and FME+CL perform similarly for the *entire range* of items hypothesized along the Precision-Recall curve. The primary difference is that FME can continue hypothesizing items even past FME+CL's threshold.

E.3 Ensembling

Results from Tables 3 and 10 show that ensembling (FME) improves error detection performance over using a single model (FML) in almost every scenario tested, at a rate several times higher than gains to underlying task performance.

We also observe a phenomenon of disproportionately high variance in model error detection performance: Table 10 quantifies the standard deviation of the former at three times the standard deviation of performance on the underlying task, and Fig. 2 shows this to be the case even when comparing models with a fixed loss. This finding persisted even when holding all hyperparameters and data constant, with only the random seed being changed.

We hypothesize that label noise in training data induces models to learn spurious correlations, which cause models to make errors in a

Method	Task Accuracy		FM Error Detection Performance			Effects of CL Overlay		
	Noisy	Clean	Precision	Recall	F1	Precision	Recall	F1
Averaged	0.88 ± 0.03	0.91 ± 0.03	0.50 ± 0.11	0.65 ± 0.03	0.56 ± 0.08	0.51 ± 0.11	0.67 ± 0.03	0.57 ± 0.07
Ensembled	0.89 ± 0.02	0.92 ± 0.03	0.56 ± 0.12	0.62 ± 0.03	0.58 ± 0.08	0.58 ± 0.11	0.65 ± 0.03	0.61 ± 0.07
Difference	+1.14%	+1.24%	+12.52%	−4.31%	+4.66%	+13.89%	−2.98%	+6.03%

Table 10: Ensembling confers gains in error detection performance disproportionate to gains in underlying task performance, across a broad range of models and hyperparameters (on TweetNLP-5, results from top three models per sweep, as measured at the fixed threshold set by CL).

structured manner (Watson et al., 2022; Jiang et al., 2022); this results in greater levels of model disagreement, with minimal impact on top-line performance. Ensembling may be disproportionately effective because it serves an added function of reducing variance caused by these low-quality features.

E.4 TAPT

We perform Task-Assisted Pretraining (TAPT; Gururangan et al., 2020) using the original hyperparameters everywhere except for the optimizer, in which we use AdamW instead of Adam for DeBERTa. We run TAPT on the all splits of the corresponding data for all datasets except Amazon Reviews, where because of its size, we use TAPT on only 50,000 data points, or 0.5% of the full dataset. After running TAPT, we then run 25 fine-tune sweeps.