NoisywikiHow: A Benchmark for Learning with Real-world Noisy Labels in Natural Language Processing

Tingting Wu[1], **Xiao Ding**[1]*, **Minji Tang**[1], **Hao Zhang**[2], **Bing Qin**[1], **Ting Liu**[1]
[1]Research Center for Social Computing and Information Retrieval
Harbin Institute of Technology, China
`{ttwu, xding, mjtang,qinb tliu}@ir.hit.edu.cn`
[2]Faculty of Computing, Harbin Institute of Technology, China
`zhh1000@hit.edu.cn`

Abstract

Large-scale datasets in the real world inevitably involve label noise. Deep models can gradually overfit noisy labels and thus degrade model generalization. To mitigate the effects of label noise, learning with noisy labels (LNL) methods are designed to achieve better generalization performance. Due to the lack of suitable datasets, previous studies have frequently employed synthetic label noise to mimic real-world label noise. However, synthetic noise is not instance-dependent, making this approximation not always effective in practice. Recent research has proposed benchmarks for learning with real-world noisy labels. However, the noise sources within may be single or fuzzy, making benchmarks different from data with heterogeneous label noises in the real world. To tackle these issues, we contribute *NoisywikiHow*, the largest NLP benchmark built with minimal supervision. Specifically, inspired by human cognition, we explicitly construct multiple sources of label noise to imitate human errors throughout the annotation, replicating real-world noise, whose corruption is affected by both ground-truth labels and instances. Moreover, we provide a variety of noise levels to support controlled experiments on noisy data, enabling us to evaluate LNL methods systematically and comprehensively. After that, we conduct extensive multi-dimensional experiments on a broad range of LNL methods, obtaining new and intriguing findings.[1]

1 Introduction

Large-scale labeled data has become indispensable in the notable success of deep neural networks (DNNs) in various domains and tasks (Russakovsky et al., 2015; Wang et al., 2019). Due to imperfect sources like crowd-sourcing and web crawling (Xiao et al., 2015; Zhang et al., 2017b; Lee et al., 2018), datasets frequently include *real-world label noise* (Chen et al., 2021), which may induce model overfitting to noisy labels and hurt the generalization of deep models (Zhang et al., 2017a; Wu et al., 2022a,b). To alleviate this issue, learning with noisy labels (LNL) methods for robustly training deep models have been studied extensively.

Input	**Output**
(a) Take prescription weight loss medications.	Losing Weight
(b) Check calories on food packaging.	
(c) Include cultural and ethnic foods in your plan.	
(d) Talk about food differently.	

Table 1: Instances (a)–(d) depict examples of our task. **Input**: a procedural event. **Output**: a plausible intention toward that event.

Due to the lack of appropriate benchmarks, previous research often studied synthetic label noise to simulate real-world label noise (Zhang et al., 2018; Lukasik et al., 2020). As a general and realistic noise, real-world noise may have several noise sources (i.e., be *heterogeneous*) (Northcutt et al., 2021) and be *instance-dependent* (i.e., $P(\tilde{y}|y, x)$, where the probability of an instance being assigned to the incorrect label $\tilde{y}$ depends on the original ground-truth label y and data x) (Han et al., 2021; Song et al., 2022). However, synthetic noise is generated from an artificial distribution and is thus *instance-independent* (i.e., $P(\tilde{y}|y)$), which may not always work well in practice.

Recently, various benchmarks for learning with real-world noisy labels have been proposed across fields like computer vision (CV) (Li et al., 2017), audio signal processing (ASP) (Gemmeke et al., 2017), and natural language processing (NLP) (Hedderich et al., 2021). To fully evaluate robust learning methods with real-world label noise, benchmarks should be as close to real-world scenarios as possible. Meanwhile, controlled ex-

[*] Corresponding author.

[1] The dataset is publicly available at `https://github.com/tangminji/NoisywikiHow`.

Findings of the Association for Computational Linguistics: ACL 2023, pages 4856–4873
July 9-14, 2023 ©2023 Association for Computational Linguistics

periments are encouraged to verify whether LNL methods can remain effective over a wide range of noise levels (Jiang et al., 2020). Nevertheless, the noise levels in most datasets are fixed and unknown, resulting in uncontrolled label noise (Fonseca et al., 2019a; Song et al., 2019). Moreover, the noise therein often comes from the same or ambiguous sources (Li et al., 2017; Jiang et al., 2020), which conflicts with the heterogeneous characteristics of real-world noise. These problems prevent a better understanding of LNL methods.

To bridge this gap, we present NoisywikiHow, a new NLP benchmark for evaluating LNL methods focusing on the intention identification task. Intention identification promotes numerous downstream natural language understanding tasks, from commonsense reasoning (Sap et al., 2019) to dialogue systems (Pepe et al., 2022). Additionally, the complexity of the task (total of 158 categories) facilitates a deeper investigation of the efficacy of LNL approaches. The task form is shown in Table 1.

To make the benchmark more representative of real-world scenarios, we propose a practical assumption: *Real-world label noise in a dataset is mainly induced by human errors, regardless of whether the dataset's construction is automated or crowd-sourced.* Existing psychological and cognitive evidence further supports our hypothesis. It shows that different annotators have different preferences and biases (Beigman and Klebanov, 2009; Burghardt et al., 2018), which means human labeling errors typically result from multiple noise sources. Furthermore, humans may make random labeling errors due to random attention slips. But they are more likely to produce label noise when labeling hard cases (Klebanov et al., 2008) (i.e., noise is instance-dependent), such as instance (c) in Table 1.

Motivated by this human cognition, we first collect data from the wikiHow website,[2] which contains a collection of professionally edited how-to guideline articles, providing a vast quantity of clean scripts and corresponding categories for free to help achieve controlled experiments and ensure benchmark quality. After that, we explicitly inject a variety of noise sources into clean data to replicate human annotation errors, thus introducing real-world label noise into the benchmark. Notably, training samples in our benchmark exhibit a long-tailed class distribution, which is in line with the facts, i.e., data in real-world applications is heavily imbalanced (Van Horn et al., 2018; Liu et al., 2019b). Besides, we achieve minimal human supervision by using a series of automated labeling procedures, saving lots of time and human effort.

To evaluate NoisywikiHow, we carry out extensive experimentation across various model architectures and noise sources, execute plentiful LNL methods on our benchmark, compare the more realistic real-world noise with the extensively studied synthetic noise, and investigate a case study and long-tailed distribution characteristics.

2 Related Work

2.1 Datasets with real-world noisy labels

In early studies of the LNL problem, due to a lack of appropriate benchmarks, synthetic noise was often used to reflect noise in the real world and assess the effectiveness of methods (Han et al., 2018b; Zhang et al., 2018). However, unlike real-world noise, synthetic noise follows an idealized artificial distribution, which leads to inaccurate approximations and inadequate evaluations.

Recent studies have proposed numerous datasets with real-world noisy labels. Table 2 depicts a comparison of existing real-world noisy datasets for evaluating LNL methods in CV, ASP, and NLP. As shown in Table 2, most datasets fail to perform controlled experiments on real-world label noise and cannot be used to study DNNs across different noise levels (Fonseca et al., 2019a,b).

A few benchmarks with controlled label noise, such as NoisyNER (Hedderich et al., 2021) and Red MiniImageNet (Jiang et al., 2020), were produced. However, the noise source in their datasets may be vague. Furthermore, NoisyNER focuses on the named entity recognition task in NLP. Though seven noisy label sets are provided, it is challenging to determine the precise noise level of each label set because a sentence-level instance has numerous word-level labels. Besides, Red MiniImageNet relies heavily on careful human annotation and follows a balanced class distribution, which diverges from real-world application scenarios. In this paper, we publish NoisywikiHow to solve the above limitations. As shown in Table 2, to the best of our knowledge, NoisywikiHow is the largest NLP benchmark for assessing LNL methods.

[2] https://www.wikihow.com

Dataset	Classes	Distribution	Controlled	Human Annotation	Size
CV					
Food-101N (Lee et al., 2018)	101	Balanced	No	No	367K
Animal-10N (Song et al., 2019)	10	Balanced	No	Yes	55K
Red MiniImageNet (Jiang et al., 2020)	100	Balanced	Yes	Yes	55K
Red Stanford Cars (Jiang et al., 2020)	196	Balanced	Yes	Yes	16.1K
Clothing1M (Xiao et al., 2015)	14	Imbalanced	No	No	1M
WebVision (Li et al., 2017)	1K	Long-tailed	No	No	2.4M
ASP					
AudioSet (Gemmeke et al., 2017)	527	Long-tailed	No	Yes	2M
FSDnoisy18K (Fonseca et al., 2019a)	20	Imbalanced	No	No	18.5K
FSDKaggle2019 (Fonseca et al., 2019b)	80	Balanced	No	No	29.2K
NLP					
NoisyNER (Hedderich et al., 2021)	4	Long-tailed	Yes	No	14.8K
NoisywikiHow	**158**	**Long-tailed**	**Yes**	**No**	**89K**

Table 2: Comparison between our benchmark and other datasets.

2.2 Intention identification

Intention identification is critical to many applications (Huang et al., 2016; Sap et al., 2019). Therefore, ensuring task reliability is essential. Some previous work formulates intention identification as an *event process typing* task. Given a sequence of events, the model is designed to understand the overall goal of the event process in terms of an action and an object (Chen et al., 2020; Pepe et al., 2022). In other studies, intention identification is modeled as a *sentence classification* task (Zhang et al., 2020a,b). When given a procedural event, the system predicts its intention in a 4-choose-1 multiple-choice format. However, none of these studies deal with task reliability. By building NoisywikiHow, we make a preliminary exploration of task reliability (i.e., model performance under label noise). Following Zhang et al. (2020b), we model intention identification as a sentence classification task. The difference is that our benchmark (including 158 labels) is analogous to the *retrieval task* in a more practical and challenging way.

3 NoisywikiHow Dataset

3.1 Data Collection

We construct NoisywikiHow by crawling how-to articles from the wikiHow website. Detailed crawling strategies and related statistics are in Appendix A.1. We define the **input** as a procedural event, i.e., the header of a paragraph in a wikiHow article (e.g., *Talk about food differently* in Table 1), and the **output** as the intention of the event, namely the category of this article (e.g., *Losing Weight* in Table 1). Note that categories present a hierarchy (e.g., *Health* $\gg$ *Nutrition and Food Health* $\gg$ *Weight Management* $\gg$ *Losing Weight*), and we select the category with the finest granularity as the label.

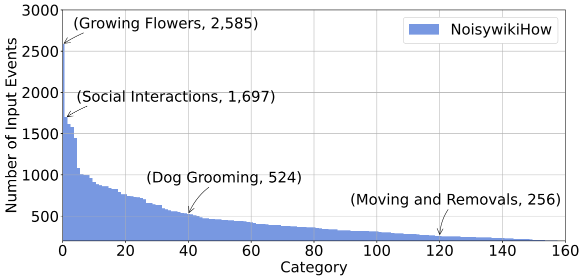


Figure 1: Number of events per category of the training set of NoisywikiHow.

3.2 Data Cleaning

Similar to Jiang et al. (2020) and Hedderich et al. (2021), we realize controlled label noise by injecting various amounts of noise into clean data. However, the data collection process introduces a lot of low-quality or irrelevant data. As a result, we develop a data cleaning procedure to remove bad data and facilitate the target task from two aspects: (1) input filtering and (2) label filtering. Regarding input filtering, we first devise four automatic filters and execute them sequentially to remove low-quality or ambiguous data.

- **Sample Length Filter** intends to retain instances with more informative and complete semantic information by filtering excessively short or long data.
- **Format Normalization** is to standardize in-

stances (e.g., unifying the description of "*Click Defragment Your **Hard Drive**.*" and "*Click Defragment **your hard drive**.*"), ensuring the effectiveness of subsequent strategies.

- **Deduplication** tries to eliminate redundant or ambiguous data (e.g., a procedural event corresponds to multiple intents).
- **TF-IDF Filter** attempts to preclude overly uninformative instances by calculating the TF-IDF for each token.

After that, we receive ***high-quality data*** $\mathcal{D}_h$, which follows a long-tailed class distribution with limited data on tail classes, resembling the distribution in Fig. 1. We create a **Sample Size Filter** to exclude the categories with too few samples (≤ 300), ensuring an appropriate split of training, validation, and test sets.

We observe that the labels have two types, i.e., concepts defined as nominal phrases (e.g., *Nutrition and Food Health*), and event mentions defined as nominal or verbal phrases that refer to events (e.g., *Losing Weight*) (Min et al., 2020; Yu et al., 2021). Therefore, label filtering is required to retain only events, ensuring the effectiveness of intention identification. Specifically, each category is annotated by three graduate students from the NLP field and is regarded as an event if more than two annotators agree. Human annotators are asked to label 736 categories and achieve a high agreement (Fleiss-κ = 0.84) (Fleiss, 1971).

After data cleaning, we obtain ***clean data*** $\mathcal{D}$ involving 89,143 instances in 158 classes. Due to the limited space, complete filtering strategies and more details are in Appendix A.2.

3.3 Label Noise Injection

To create a benchmark of real-world noisy labels, we introduce various sources of controlled label noise into the clean data. Prior to this, we assume that ***human mistake is the primary cause of real-world label noise in a dataset***. Psychological and cognitive findings further corroborate the rationality of the assumption. It demonstrates that: (1) apparent differences between annotators result from different preferences and biases (Reidsma and op den Akker, 2008; Beigman and Klebanov, 2009; Burghardt et al., 2018), suggesting that human errors are ***heterogeneous***; (2) label noise from humans regularly affects hard cases (Klebanov et al., 2008; Klebanov and Beigman, 2009), proving that noise is ***instance-dependent***.

Heterogeneous noise sources. Based on the above preliminaries, we simulate various mistakes committed by annotators to produce real-world noise containing heterogeneous noise sources. Specifically, human errors are often induced by ambiguity, insufficient annotator expertise, and random attention slips (Beigman and Klebanov, 2009; Hollenstein et al., 2016). Motivated by this, we develop three noise sources as follows:

- Sub-categories (**SC**) under the same category (e.g., *Starting a Business* and *Running a Business*) tend to have higher semantic similarities and can be easily confused. SC depicts the noise caused by labeling ambiguous instances.
- Intents beyond the commonsense categories (**BCC**) are hard to identify (e.g., *Dog Grooming*), readily inducing noisy labels. BCC portrays a scenario annotated by a human lack of expert knowledge.
- Considering the long-tailed distribution, even a few labeling errors on tail classes can seriously affect learning of these categories. Therefore, achieving robust training on tail classes is critical. We concentrate on intents under the tail categories (**TC**), which describe the noise generated by humans randomly shifting their attention.

Then, we design a simple mapping from noise sources to classes to facilitate the subsequent injection of noise from different sources and categories. Specifically, each class is associated with a noise source, and classes under various noise sources do not overlap. This mapping can cover all categories during noise injection and determine the potential noise source for each class. Finally, we divide 158 categories into 68, 36, and 54 to correspond to the sources SC, BCC, and TC, respectively. More details about the mapping can be found in Appendix A.3.

Injecting instance-dependent label noise. Since each noise source contains a set of categories, each of which may involve hard cases, instance-dependent label noise exists in each noise source. Note that real-world label noise always comes from an open rather than a finite category set (Wang et al., 2018). We therefore enable label noise to derive from categories other than the current label set. However, this operation changes the number of labels and impacts the target classification task. To solve this problem, when injecting label noise into an instance (x, y), we leave the label y (*output*) unchanged like Jiang et al. (2020) but replace the procedural event x (*input*) with the one ($\tilde{x}$) under the other category ($\tilde{y}$), which may not be in the existing 158 classes. Moreover, NoisywikiHow supports five noise levels (i.e., 0%, 10%, 20%, 40%, and 60%). Like Li et al. (2017) and Saxena et al. (2019), we assume that, given a specified noise level t, t is uniform across noise sources. For example, $t = 10\%$ represents that each source has roughly 10% label noise.

We further identify hard cases and inject instance-dependent noise for each noise source. Intuitively, when we mislabel an instance from (x, y) into $(\tilde{x}, y)$, if (x, y) is a hard case, the semantic representations of events x and $\tilde{x}$ should be very similar. As a result, for any (x, y), we can assess its difficulty by finding an $(\tilde{x}, \tilde{y})$ whose $\tilde{x}$ has the maximum semantic similarity with x. To identify $(\tilde{x}, \tilde{y})$, we take the following steps: (1) **Determine $\mathcal{D}_n$: the candidate set of** $(\tilde{x}, \tilde{y})$. To avoid introducing bad data or duplicate data after noise injection, we construct $\mathcal{D}_n$ as follows:

- For the sources BCC and TC, $\mathcal{D}_n = \mathcal{D}_h - \mathcal{D}$.
- For the source SC, let $\mathcal{D}_s$ be the sample set of all other sub-categories except y under the same category, and $\mathcal{D}_n = (\mathcal{D}_h - \mathcal{D}) \cap \mathcal{D}_s$.

(2) **Locate $\tilde{x}$ in $\mathcal{D}_n$.** Following Zhang et al. (2020b), we map each event to a vector representation by taking the average of the BART embeddings (Lewis et al., 2020) of the verbs. $\tilde{x}$ thus can be calculated as:

$$\tilde{x} = \arg\max_{v_{x'}} cosine(v_x, v_{x'}), (x', y') \in \mathcal{D}_n, \tag{1}$$

where $v_{(\cdot)}$ is the vector representation of an event, and $cosine(\cdot)$ denotes the cosine similarity of two vectors. For any (x, y), its difficulty can be obtained by calculating a score s_x:

$$s_x = cosine(v_x, v_{\tilde{x}}). \tag{2}$$

The larger the s_x, the harder the instance (x, y).

We inject noise into the training set $\mathcal{D}_{tr} \subset \mathcal{D}$. Given a specified noise level t (e.g., 10%), all instances in $\mathcal{D}_{tr}$ are arranged in decreasing order of s_x, with the top t of the samples in each source considered hard cases. We inject instance-dependent noise by replacing x for each hard case with $\tilde{x}$.

4 Experiments

We first present the general settings for experiments (Section 4.1). Further, we systematically evaluate our benchmark with varied model architectures (Section 4.2) and noise sources (Section 4.3). Also, we assess a broad range of LNL methods on NoisywikiHow (Section 4.4) and compare real-world noise with synthetic noise (Section 4.5). Finally, we conduct a case study (Section 4.6). In addition, we discuss the long-tailed distribution characteristics of NoisywikiHow in Appendix B.2.

Noise Level(%): 0, 10, 20, 40, 60					
Noise Sources	**Class**	**Train**	**Val**	**Test**	**Total**
SC	68	39,674	3,400	3,400	46,474
BCC	36	20,413	1,800	1,800	24,013
TC	54	13,256	2,700	2,700	18,656
Total	158	73,343	7,900	7,900	89,143

Table 3: Overview of NoisywikiHow of multiple noise sources and controlled label noise, where SC, BCC, and TC denote noise sources from sub-categories, categories beyond the commonsense, and tail categories.

4.1 Experimental settings

On our benchmark, all methods are trained on the noisy training sets[3] and evaluated on the same clean validation set to verify whether these approaches can resist label noise during training and achieve good generalization on the noise-free data. Before adding label noise, we randomly split out 15,800 instances from clean data and then equally divide them into two sets: a validation set and a test set. The remaining 73,343 instances serve as the training set, which follows a typical long-tailed class distribution and is analogous to heavily imbalanced data in real-world applications, as shown in Fig. 1. The statistics of NoisywikiHow are shown in Table 3. We cast intention identification as a classification problem. We exploit the cross-entropy loss for training models and use Top-1 accuracy and Top-5 accuracy as the evaluation metrics.

4.2 Comparison of Model Architectures

Baselines: We first evaluate the performance of different model architectures under varying levels of real-world label noise. Regarding the model architectures, we use seven state-of-the-art (SOTA) pre-trained language models, including BERT (Devlin et al., 2019), XLNet (Yang et al., 2019), RoBERTa (Liu et al., 2019a), GPT2 (Radford et al., 2019), ALBERT (Lan et al., 2020), T5 (Raffel et al.,

[3] Synthetic noise and various noise sources under real-world noise correspond to diverse noisy training sets.

Method	Noise Level				
	0%	10%	20%	40%	60%
	Top-1(Top-5)	Top-1(Top-5)	Top-1(Top-5)	Top-1(Top-5)	Top-1(Top-5)
BERT (Devlin et al., 2019)	60.29(83.53)	58.86(83.82)	57.42(82.57)	52.91(79.84)	48.20(75.37)
XLNET (Yang et al., 2019)	59.77(85.24)	60.23(85.90)	58.25(84.29)	53.74(81.73)	50.23(**79.44**)
RoBERTa (Liu et al., 2019a)	60.59(85.10)	59.65(84.16)	57.77(83.77)	54.18(81.56)	**50.85**(78.87)
GPT2 (Radford et al., 2019)	59.84(85.39)	58.35(84.90)	57.0(83.94)	52.71(80.81)	48.25(78.08)
ALBERT (Lan et al., 2020)	55.13(80.80)	56.21(82.15)	53.68(80.52)	49.93(78.44)	44.81(74.41)
T5 (Raffel et al., 2020)	58.35(83.63)	56.87(83.03)	56.19(82.20)	52.29(79.94)	47.47(77.39)
BART (Lewis et al., 2020)	**61.72(86.90)**	**60.28(85.92)**	**58.94(84.67)**	**54.57(82.38)**	49.75(78.84)

Table 4: Top-1 (Top-5) classification accuracy (%) of pre-trained language models on the NoisywikiHow test set under different levels of real-world label noise. Top-1 results are in bold.

2020), and BART (Lewis et al., 2020). We finetune each model for 10 epochs with batch size 64, learning rate 3e-5. These hyperparameters remain unchanged in subsequent experiments unless indicated otherwise. In this paper, we conduct all experiments utilizing the base-sized version of the pre-trained language models. Besides, due to long output sequences in partial categories, we adopt beam search (Sutskever et al., 2014) in T5, with a beam width of 5 and a length penalty of $\alpha = 1.0$.

Results: As shown in Table 4, the Top-1 accuracies of SOTA pre-trained language models on our benchmark are generally not high, and an increase in noise levels can lead to considerable performance degradation for a given model, demonstrating the challenge of the NoisywikiHow dataset.

In Table 4, different architectures are representative of diverse capacities. For example, RoBERTa and XLNet consistently outperform ALBERT under different noise levels. In addition, we observe that BART achieves the best performance among these SOTA models under a majority of noise levels, regardless of Top-1 or Top-5 classification accuracy. This is mainly because a better *denoising* objective (i.e., *masked language modeling*) is used during pre-training of BART. In pre-training, BART gains better denoising ability by corrupting text with an arbitrary noise function (thus making the noise more flexible) and learning to reconstruct the original text. In the following, we use the BART model as the ***base model***.

4.3 Effects of Distinct Noise Sources

We further explore the characteristics of different noise sources. To this end, we pick the same model (i.e., the base model) and separately validate the performances on individual noise sources under the same noise level. For convenience, we denote noise-free data by *correct samples* and data with label noise by *incorrect samples*.

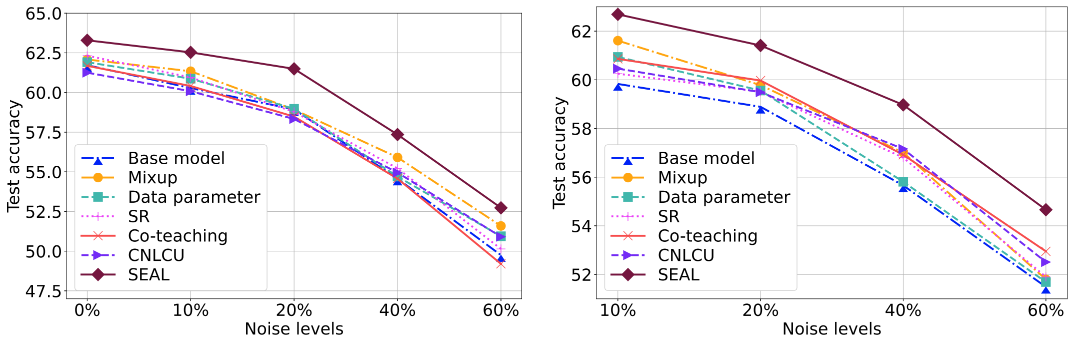
Noise Sources	Top-1	Top-5
SC+BCC+TC	60.28	85.92
SC	60.14	85.49
BCC	59.65	85.39
TC	57.99	84.37

Table 5: Test accuracy (%) of the base model under distinct noise sources with 10% label noise, where SC+BCC+TC denotes the default NoisywikiHow with a mixture of noise sources.

Results: Table 5 shows the results of the base model under four different noise sources with 10% label noise. As shown in Table 5, there exists an evident gap between the results under noise source TC and those in other conditions. The label noise from noise source TC is more difficult to mitigate than others at the same noise level, mainly due to the limited data on tail categories. When all noisy labels are derived from TC, fewer correct samples are left, leading to inadequate model training and degradation of model performance. It indicates that resisting label noise from different sources may have varying difficulty levels, although the noises in these sources are all real-world label noise. Additional details are provided in Appendix B.1.

4.4 Effectiveness of Different LNL methods

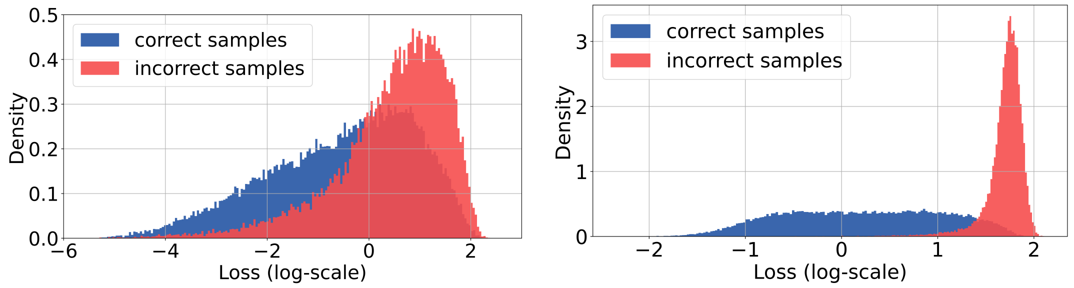
Baselines: We perform an extensive evaluation of the existing LNL methods on our benchmark. Seven representative baselines are involved for comparison: (1) ***Base model***, which finetunes the BART model with no extra LNL methods; (2) ***Mixup*** (Zhang et al., 2018), which mitigates memorization of noisy labels by DNNs regularization, i.e., introducing a data-agnostic data augmentation routine; (3) ***Data Parameter*** (Saxena et al., 2019),



(a) Real-world label noise.

(b) Synthetic label noise

Figure 2: Test accuracy (Top-1) of representative LNL methods trained with controlled label noise.



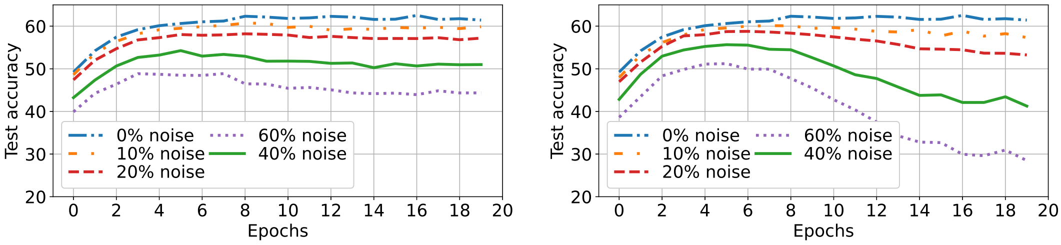
(a) Real-world label noise.

(b) Synthetic label noise

Figure 3: Training loss distributions of correct samples and incorrect ones at the 4-th epoch with 40% label noise.

which equips learnable parameters to help DNNs generalize better via learning from easier instances first; (4) ***SR*** (Zhou et al., 2021), which introduces the sparse regularization strategy, making any loss robust to noisy labels conforming to the specified assumption; (5) ***Co-teaching*** (Han et al., 2018b), which combats noisy labels by training two networks, and each network aims to teach the other one with clean data, i.e., the instances with small-loss; (6) ***CNLCU*** (Xia et al., 2022), which considers the uncertainty of loss estimation to refine correct sample selection; (7) ***SEAL*** (Chen et al., 2021), which provides instance-dependent label correction to resist real-world noise. Complete experimental results and unique hyperparameters for each noise level for each baseline are in Tables 8 and 9 in the Appendix.

Results: As Fig. 2(a) shows, Mixup outperforms the base model with limited performance improvement. It is because ***Mixup fails to consider the specialty of real-world label noise*** and improves generalization with a generic regularization-based method. The performance of Data Parameter is comparable to or slightly better than the base model under different noise levels. Although Data Parameter models the situation that instances within a class have different difficulty levels, it assumes *small-loss training samples as correct samples and splits correct and incorrect samples via a loss-based separation*. However, as shown in Fig. 3(a), loss distributions of correct and incorrect data overlap closely in the real-world label noise, making ***Data Parameter has no advantage under real-world label noise***. Similarly, Co-teaching and CNLCU fulfill sample selection following the same assumptions. They perform worse than the base model, with the exception of individual noise levels. It implies that ***Co-teaching and CNLCU are inapplicable to the heterogeneous and instance-dependent label noise***. SR precedes the base model only at certain noise levels. This is because SR guarantees noise tolerance if and only if the label noise satisfies the instance-independent condition, which is inconsistent with noise in the real world. Hence, ***the validity of SR is not ensured on our benchmark***. SEAL consistently outperforms the base model by a large margin on all noise levels, as SEAL provides instance-dependent label correction to combat real-world noise. However, during the correction, SEAL retrains the classifier using



(a) Real-world label noise

(b) Synthetic label noise

Figure 4: Test accuracy of the base model trained with controlled label noise.

Noise Sources	Incorrect sample	Noisy label	Ground-truth label (unobservable)
SC	(a) Rinse off the paste using warm water.	Coloring Hair	Making Skin Look Lighter
BCC	(b) Mow your lawn and the leaves.	Lawn Care	Cleaning up Garden
	(c) Avoid over-fertilizing your tree.	Growing Trees and Shrubs	Growing Fruit
TC	(d) Give yourself a span of time to mourn.	Domestic Violence	Rebuilding Life After Divorce
	(e) Place the bananas on a wire rack.	Steaming Food	Food Preservation Techniques

Table 6: Five incorrect instances from three different noise sources in the NoisywikiHow dataset.

the averaged soft labels, introducing excessive computational overhead.

4.5 Real-world Noise vs. Synthetic Noise

Aside from the real-world label noise, synthetic label noise is one of the most widely studied label noises (Patrini et al., 2017; Wang et al., 2018; Reeve and Kabán, 2019). Unlike real-world noise, which is widespread in real applications, synthetic noise does not exist but is generated from artificial distributions. We further examine the differences between the two label noises. In this paper, synthetic label noise is implemented with symmetric label noise (Han et al., 2018a; Charoenphakdee et al., 2019) (the most common synthetic noise), assuming each label has the same probability of flipping to any other class. We build the dataset of controlled synthetic label noise by injecting a series of synthetic label noises into clean data in a controlled manner (i.e., 10%, 20%, 40%, and 60% noise levels). We pick the same baselines as in Section 4.4. More details are in Tables 10 and 11 in the Appendix.

Results: As shown in Fig. 2(b), SEAL and Mixup consistently outperform the base model, showing their advantages in combating synthetic label noise. Unlike the real-world label noise, SR is effective for the synthetic label noise and achieves improvement over the base model regardless of the noise levels since the synthetic label noise meets the instance-independent condition. Besides, Co-teaching, Data Parameter, and CNLCU improve the base model by an apparent margin under the synthetic label noise. In this case, as shown in Fig. 3(b), the loss distributions of correct and incorrect samples can be well split, allowing loss-based separation to work well.

We discover that few LNL methods can effectively resist both real-world and synthetic noises simultaneously, highlighting the imperative of benchmark construction. Many LNL approaches can mitigate the synthetic but not real-world label noise. It is because synthetic noise is generated from artificial distributions to approximate real-world noise. The mislabeled probability is independent of each instance under synthetic noise but dependent on distinct instances under real-world noise, which makes complex modeling of the latter. Thus, our benchmark contributes to a more systematic and comprehensive assessment of LNL methods. Further, since most LNL method evaluation datasets focus on the CV and ASP, our NLP benchmark facilitates the modal integrity of the existing datasets.

We also contrast the performance of the base model trained for 20 epochs under real-world label noise and synthetic label noise. In Fig. 4, as the running epochs and noise levels increase, the test accuracy curve with the real-world noise (Fig. 4(a)) is much flatter than that with the synthetic noise (Fig. 4(b)) at the same noise level (e.g., with 40% and 60% noise). It demonstrates that the model generalizes much better under real-world noise than synthetic noise of the same noise level.

4.6 Case Study

We construct a benchmark encompassing real-world noise involving multiple noise sources with minimal human supervision, which is analogous to human errors during annotation. To observe the dataset more clearly and intuitively, we randomly select five incorrect instances (i.e., samples with noisy labels) across multiple noise sources. As indicated in Table 6, we find it difficult to determine whether the sample contains noise. On the other hand, for any sample, the noise label and the respective ground-truth label are overly similar, making it challenging to distinguish one from another.

5 Conclusion

In this paper, we study the problem of learning with noisy labels and establish an NLP benchmark called NoisywikiHow with minimal human supervision, which contains more than 89K procedural events with heterogeneous and controlled real-world label noise. Experimental results reveal several new findings. (1) Some widely accepted LNL methods are not always impactful, especially with real-world label noise. (2) Different noise sources may have varying difficulties resisting label noise, although they are all from real-world noise. (3) Few LNL methods can effectively combat real-world noise and synthetic noise at the same time. (4) The model trained under the real-world label noise has better generalization performance.

Limitations

In this paper, we simplify intention identification into a sentence classification task, i.e., exploiting a specific procedural event in an event process to predict the intention of the whole event process. A more realistic way to model this task is to enter the entire event process rather than a single event. We will go into more detail about this type of task in future work.

Ethics Statement

This work presents NoisywikiHow, a free and open dataset for the research community to study learning with noisy labels. Since the data in NoisywikiHow is constructed based on the wikiHow website, which is free and open for academic usage, there is no privacy issue. We declare that all information in this paper has been obtained and presented following the ACL Ethics Policy. As required by these rules and conduct, we have fully cited and referenced all material and results that are not original to this work.

Acknowledgements

We thank the anonymous reviewers for their constructive comments, and gratefully acknowledge the support of the Technological Innovation "2030 Megaproject" - New Generation Artificial Intelligence of China (2020AAA0106501), the National Natural Science Foundation of China (U22B2059, 62176079, 62106061), and the Natural Science Foundation of Heilongjiang Province (YQ2022F005).

References

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

James Bergstra, Daniel Yamins, and David Cox. 2013. Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. In *International conference on machine learning*, pages 115–123. PMLR.

Keith Burghardt, Tad Hogg, and Kristina Lerman. 2018. Quantifying the impact of cognitive biases in question-answering systems. In *Twelfth International AAAI Conference on Web and Social Media*.

Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. 2019. Learning imbalanced datasets with label-distribution-aware margin loss. *Advances in neural information processing systems*, 32.

Nontawat Charoenphakdee, Jongyeong Lee, and Masashi Sugiyama. 2019. On symmetric losses for learning from corrupted labels. In *International Conference on Machine Learning*, pages 961–970. PMLR.

Muhao Chen, Hongming Zhang, Haoyu Wang, and Dan Roth. 2020. What are you trying to do? semantic typing of event processes. In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 531–542.

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics*.

Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378.

Eduardo Fonseca, Manoj Plakal, Daniel PW Ellis, Frederic Font, Xavier Favory, and Xavier Serra. 2019a. Learning sound event classifiers from web audio with noisy labels. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 21–25. IEEE.

Eduardo Fonseca, Manoj Plakal, Frederic Font, Daniel PW Ellis, and Xavier Serra. 2019b. Audio tagging with noisy labels and minimal supervision. *arXiv preprint arXiv:1906.02975*.

Jort F Gemmeke, Daniel PW Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R Channing Moore, Manoj Plakal, and Marvin Ritter. 2017. Audio set: An ontology and human-labeled dataset for audio events. In *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 776–780. IEEE.

Bo Han, Jiangchao Yao, Gang Niu, Mingyuan Zhou, Ivor Tsang, Ya Zhang, and Masashi Sugiyama. 2018a. Masking: A new perspective of noisy supervision. *Advances in neural information processing systems*, 31.

Bo Han, Quanming Yao, Tongliang Liu, Gang Niu, Ivor W Tsang, James T Kwok, and Masashi Sugiyama. 2021. A survey of label-noise representation learning: Past, present and future.

Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor W. Tsang, and Masashi Sugiyama. 2018b. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *Neural Information Processing Systems*.

Michael A Hedderich, Dawei Zhu, and Dietrich Klakow. 2021. Analysing the noise model error for realistic noisy label data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 7675–7684.

Nora Hollenstein, Nathan Schneider, and Bonnie Webber. 2016. Inconsistency detection in semantic annotation. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3986–3990.

Matthew Honnibal and Ines Montani. 2017. spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. *To appear*, 7(1):411–420.

Lifu Huang, Taylor Cassidy, Xiaocheng Feng, Heng Ji, Clare Voss, Jiawei Han, and Avirup Sil. 2016. Liberal event extraction and event schema induction. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 258–268.

Lu Jiang, Di Huang, Mason Liu, and Weilong Yang. 2020. Beyond synthetic noise: Deep learning on controlled noisy labels. In *International Conference on Machine Learning*, pages 4804–4815. PMLR.

Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. 2020. Decoupling representation and classifier for long-tailed recognition. In *Eighth International Conference on Learning Representations (ICLR)*.

Beata Beigman Klebanov and Eyal Beigman. 2009. Squibs: From annotator agreement to noise models. *Computational Linguistics*, 35(4):495–503.

Beata Beigman Klebanov, Eyal Beigman, and Daniel Diermeier. 2008. Analyzing disagreements. In *Coling 2008: Proceedings of the workshop on Human Judgements in Computational Linguistics*, pages 2–7.

Dimitrios Kouzis-Loukas. 2016. *Learning Scrapy*. Packt Publishing Ltd.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*.

Kuang-Huei Lee, Xiaodong He, Lei Zhang, and Linjun Yang. 2018. Cleannet: Transfer learning for scalable image classifier training with label noise. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5447–5456.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.

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

Hugo Liu and Push Singh. 2004. Conceptnet—a practical commonsense reasoning tool-kit. *BT technology journal*, 22(4):211–226.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019a. Roberta: A robustly optimized bert pretraining approach. *arXiv: Computation and Language*.

Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X Yu. 2019b. Large-scale long-tailed recognition in an open world. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2537–2546.

Michal Lukasik, Srinadh Bhojanapalli, Aditya Menon, and Sanjiv Kumar. 2020. Does label smoothing mitigate label noise? In *International Conference on Machine Learning*, pages 6448–6458. PMLR.

Bonan Min, Yee Seng Chan, and Lingjun Zhao. 2020. Towards few-shot event mention retrieval: An evaluation framework and a siamese network approach. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 1747–1752.

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

Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. 2017. Making deep neural networks robust to label noise: A loss correction approach. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1944–1952.

Sveva Pepe, Edoardo Barba, Rexhina Blloshmi, and Roberto Navigli. 2022. Steps: Semantic typing of event processes with a sequence-to-sequence approach.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:1–67.

Henry Reeve and Ata Kabán. 2019. Fast rates for a knn classifier robust to unknown asymmetric label noise. In *International Conference on Machine Learning*, pages 5401–5409. PMLR.

Dennis Reidsma and Rieks op den Akker. 2008. Exploiting 'subjective'annotations. In *Coling 2008: Proceedings of the workshop on Human Judgements in Computational Linguistics*, pages 8–16.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. 2015. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252.

Maarten Sap, Ronan Le Bras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A Smith, and Yejin Choi. 2019. Atomic: An atlas of machine commonsense for if-then reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3027–3035.

Shreyas Saxena, Oncel Tuzel, and Dennis DeCoste. 2019. Data parameters: A new family of parameters for learning a differentiable curriculum. *Advances in Neural Information Processing Systems*, 32.

Hwanjun Song, Minseok Kim, and Jae-Gil Lee. 2019. Selfie: Refurbishing unclean samples for robust deep learning. In *International Conference on Machine Learning*, pages 5907–5915. PMLR.

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*.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27.

Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. 2018. The inaturalist species classification and detection dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8769–8778.

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

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

Tingting Wu, Xiao Ding, Minji Tang, Hao Zhang, Bing Qin, and Ting Liu. 2022a. Stgn: an implicit regularization method for learning with noisy labels in natural language processing. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, page 7587–7598.

Tingting Wu, Xiao Ding, Hao Zhang, Jinglong Gao, Li Du, Bing Qin, and Ting Liu. 2022b. Discrimloss: A universal loss for hard samples and incorrect samples discrimination. *arXiv preprint arXiv:2208.09884*.

Xiaobo Xia, Tongliang Liu, Bo Han, Mingming Gong, Jun Yu, Gang Niu, and Masashi Sugiyama. 2022. Sample selection with uncertainty of losses for learning with noisy labels. In *International Conference on Learning Representations*.

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

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Neural Information Processing Systems*.

Xiaodong Yu, Wenpeng Yin, Nitish Gupta, and Dan Roth. 2021. Event linking: Grounding event mentions to wikipedia. *arXiv preprint arXiv:2112.07888*.

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

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. 2018. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*.

Li Zhang, Qing Lyu, and Chris Callison-Burch. 2020a. Intent detection with wikihow. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 328–333.

Li Zhang, Qing Lyu, and Chris Callison-Burch. 2020b. Reasoning about goals, steps, and temporal ordering with wikihow. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4630–4639.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D Manning. 2017b. Position-aware attention and supervised data improve slot filling. In *Conference on Empirical Methods in Natural Language Processing*.

Boyan Zhou, Quan Cui, Xiu-Shen Wei, and Zhao-Min Chen. 2020. Bbn: Bilateral-branch network with cumulative learning for long-tailed visual recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9719–9728.

Xiong Zhou, Xianming Liu, Chenyang Wang, Deming Zhai, Junjun Jiang, and Xiangyang Ji. 2021. Learning with noisy labels via sparse regularization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 72–81.

A Details of Dataset Construction

A.1 Crawling Strategy

According to wikiHow's crawler rules,[4] we use the crawling platform Scrapy (Kouzis-Loukas, 2016) to crawl all the articles in the 19 top-level categories (e.g., *Arts and Entertainment*, *Computers and Electronics*, etc.) of the latest wikiHow website, with a total of 100,623 pages (how-to articles), including 1,407,306 samples in 3,334 categories, as shown in Table 7.

A.2 Filtering Strategies

In the main paper, we apply a collection of filters to ensure low-quality instances removal, better dataset division, and task effectiveness. The details of each filter are as follows:

Sample Length Filter: We remove instances with overly short or long event descriptions or with icon information. As too-short events may be less informative, too-long depictions may exceed the length restriction of the pre-trained language model. Icons in events present rich text starting with "smallUrl" without specific semantic information and may interfere with the understanding of procedural events.

Format Normalization: We observe that some identical event descriptions would be slightly different in distinct articles (e.g., "*Click Defragment Your **H**ard **D**rive.*" and "*Click Defragment **y**our **h**ard **d**rive.*"). Prior to the deduplication procedure, we devise format standardization operations. The manipulations involve standardizing varied languages and symbols with Unidecode, stopword exclusion and lemmatization with spaCy (Honnibal and Montani, 2017), word segmentation & POS tagging by applying the model "en_core_web_sm" in spaCy and reserving events containing verbs.

Deduplication: We first apply inter-class deduplication to remove instances with labels of multiple categories. Then, we filter out repeated samples to achieve in-class deduplication. After the deduplication operation, each procedural event (i.e., event) corresponds to a unique event intent (i.e., category).

TF-IDF Filter: We exploit the TF-IDF filter to preclude events from being overly uninformative when identifying the corresponding event intent and guarantee the instances are representative. Specifically, each wikiHow article is considered a document. We calculate the TF-IDF for each token

[4] https://www.wikihow.com/robots.txt

and preserve only the events containing keywords. In this context, *keywords* refer to tokens whose TF-IDF values are in the top 10% in decreasing order. Each article includes a minimum of 3 and a maximum of 10 keywords.

Label Filter: We filter labels by manual annotation to retain categories depicting only events. For human labeling, we used three graduate students from the NLP field. They were educated for two hours about annotation strategy before the labeling process. Specifically, we use Min et al. (2020)'s and Yu et al. (2021)'s definitions of *event mention* (i.e., an event with surrounding context (text)) as guidelines for annotating events. In addition, categories exhibit a hierarchical structure. Typically, the descriptions of the upper categories are relatively general and vague (e.g., *Cleaning*), while the more fine-grained categories have more specific intentions (e.g., *Kitchen Cleaning*, *Cleaning Metals*). Accordingly, we label the category with the finest granularity as an event except for two cases.

- If a candidate category has a broad intent meaning (e.g., ***Selling*** in *Finance and Business ≫ Managing Your Money ≫ Making Money ≫ Selling*), it will not be considered an event.

- If it is difficult to distinguish semantically between two candidate categories, the category with the larger sample size is designated as an event. For example, in hierarchical categories (*Hobbies and Crafts ≫ Crafts ≫ Needlework ≫ Knitting and Crochet ≫ Crochet ≫ Crochet Stitches*), we label ***Crochet*** (with 1,263 samples) as an event rather than ***Crocheet Stitches*** (with 445 samples).

This annotation strategy facilitates the balance between definite event intent and sample size. Sample size and class info reserved after data cleaning are provided in Table 7.

A.3 Mapping from Noise Sources to Classes

In the main paper, we briefly present the correspondence between noise sources and task categories. In particular, we first define 54 tail categories, each containing no more than 400 samples. Following that, we draw on the discussion of commonsense knowledge in Liu and Singh (2004)[5] and use it as a guideline for labeling categories beyond commonsense. We define the overall 45 categories beyond commonsense by asking three annotators to label 158 categories as commonsense or not, fulfilling a high agreement (Fleiss-κ = 0.88). To ensure that the label sets under different noise sources do not overlap, we remove 9 categories also appearing in tail categories from the 45 categories and eventually receive 36 categories beyond commonsense. Lastly, the remaining 68 of the 158 classes are designated as the noise source SC.

Operation		Class	Size
Crawling		3,334	1,407,306
Input Filtering	Sample Length Filter	3,334	777,135
	Format Normalization		
	Deduplication		
	TF-IDF Filter		
	Sample Size Filter	736	412,080
Label Filtering		158	89,143

Table 7: Statistics of reserved valid sample size and classes after different operations.

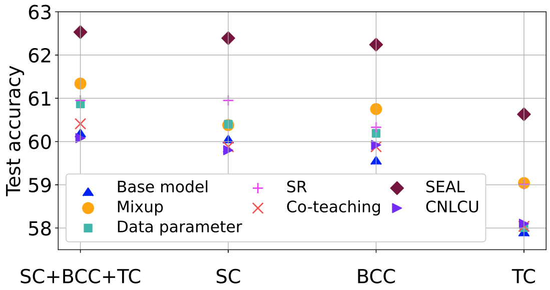


Figure 5: Test accuracy (%) of typical LNL methods under distinct noise sources with 10% label noise.

B Experiments Details

For each experimental dimension, we refine the hyperparameters for every baseline across different noise levels. Optimal hyperparameters are obtained by using a popular hyperparameter optimization tool *Hyperopt* (Bergstra et al., 2013).

B.1 Effects of Distinct Noise Sources

We examine the base model's performance under four different noise sources. In addition, Fig. 5 further compares the efficacy of typical LNL methods under various noise sources. We discover that regardless of the method employed, they are all less effective in reducing the effect of the noise source TC, further confirming our point of view.

[5] See Section 1.1 for more details.

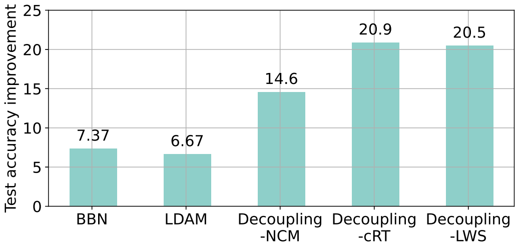


Figure 6: Performance improvements under different long-tailed learning methods in original papers.

B.2 Long-tailed Distribution Properties

Baselines: Our training set follows a typical long-tailed class distribution akin to that in the real world. However, DNNs can be readily biased towards dominant classes with massive training data, triggering poor model performance on tail classes with limited data. This problem inspires large numbers of long-tailed learning studies. To fully explore the characteristics of the NoisywikiHow dataset, we select five long-tailed learning methods in three classical categories as baselines: (1) ***BBN*** (Zhou et al., 2020), which applies a resampling strategy to sample more tail-class samples for improving tail-class performance; (2) ***LDAM*** (Cao et al., 2019), which rebalances classes by designing an effective loss and training schedule; (3) ***Decoupling*** (Kang et al., 2020), which decouples the learning procedure (including three baselines: ***Decoupling-NCM***, ***Decoupling-cRT***, and ***Decoupling-LWS***) to understand how the long-tailed recognition ability is achieved. Complete experimental results of long-tailed learning methods are shown in Table 12. We also demonstrate the settings of optimal hyperparameters in Table 13.

Results: We focus on the relative performance boost with various baselines in original papers and that on Noisywikihow. In Fig. 6, we find that all baselines evaluated on the CV datasets can address the long-tailed problem properly and achieve a significant test accuracy boost (7.37%–20.9%) in the original papers. However, as shown in Fig. 7, the performance improvements across varied noise levels on our NLP benchmark are limited, with some methods not exceeding the base model (-0.07%–2.56%).

Experimental results indicate that the effectiveness of long-tailed learning methods needs to be examined on datasets with different modals. Moreover, although the base model obtains performance degradation with the increase in the noise level, the effectiveness of each long-tailed learning method is not significantly affected by the noise level variation. The main reason is that the test accuracy we report is the best peak accuracy, producing an effect similar to the early stop and thus preventing the model from overfitting label noise.

Method	Noise Level				
	0%	10%	20%	40%	60%
	Top-1(Top-5)	Top-1(Top-5)	Top-1(Top-5)	Top-1(Top-5)	Top-1(Top-5)
Base model (Lewis et al., 2020)	61.72(86.90)	60.28(85.92)	58.94(84.67)	54.57(82.38)	49.75(78.84)
Mixup (Zhang et al., 2018)	62.08(86.99)	61.34(86.61)	58.92(85.76)	55.91(83.49)	51.59(81.09)
Data Parameter (Saxena et al., 2019)	61.91(86.54)	60.87(86.24)	58.97(85.56)	54.70(82.05)	50.94(79.70)
SR (Zhou et al., 2021)	62.32(87.35)	60.95(86.23)	58.78(86.09)	55.22(82.81)	50.14(79.70)
Co-teaching (Han et al., 2018b)	61.68(87.04)	60.41(86.11)	58.48(83.99)	54.57(81.58)	49.20(77.06)
CNLCU (Xia et al., 2022)	61.25(86.67)	60.08(85.25)	58.33(83.52)	54.95(81.20)	50.91(78.08)
SEAL (Chen et al., 2021)	**63.29(87.65)**	**62.53(87.27)**	**61.49(86.57)**	**57.35(84.41)**	**52.73(81.56)**

Table 8: Top-1 (Top-5) classification accuracy (%) of representative LNL methods on the test set of NoisywikiHow under different noise levels. Top-1 results are in bold.

Method	Optimal Hyperparameters Settings
Mixup (Zhang et al., 2018)	$\alpha = 1$
Data Parameter (Saxena et al., 2019)	lr_inst_param=0.2, wd_inst_param=0.0
SR (Zhou et al., 2021)	$\tau = 0.05, \lambda_0 = 0$, epochs=20
Co-teaching (Han et al., 2018b)	$T_k = 8, \tau = \epsilon$ (ϵ is the noise level)
CNLCU (Xia et al., 2022)	$T_k = 8, \tau_{min} = 0.3$, fixed-length time intervals=5
SEAL (Chen et al., 2021)	Number of iterations=4

Table 9: Optimal hyperparameter settings for different controlled real-world label noise on NoisywikiHow.

Method	Noise Level			
	10%	20%	40%	60%
	Top-1(Top-5)	Top-1(Top-5)	Top-1(Top-5)	Top-1(Top-5)
Base model	59.83(85.66)	58.89(84.97)	55.66(82.03)	51.49(78.29)
Mixup (Zhang et al., 2018)	61.61(86.40)	59.78(85.57)	57.01(83.20)	51.80(78.97)
Data Parameter (Saxena et al., 2019)	60.94(85.74)	59.56(85.39)	55.81(82.41)	51.69(78.73)
SR (Zhou et al., 2021)	60.25(82.35)	59.51(81.54)	56.80(79.71)	51.90(77.38)
Co-teaching (Han et al., 2018b)	60.86(86.06)	59.97(85.16)	56.92(82.99)	52.95(79.85)
CNLCU (Xia et al., 2022)	60.46(85.84)	59.49(84.92)	57.16(83.05)	52.51(78.28)
SEAL (Chen et al., 2021)	**62.69(87.66)**	**61.41(86.99)**	**58.97(84.77)**	**54.66(80.92)**

Table 10: Top-1 (Top-5) test accuracy (%) of representative LNL methods with controlled synthetic label noise.

Method	Optimal Hyperparameters Settings
Mixup (Zhang et al., 2018)	$\alpha = 1$
Data Parameter (Saxena et al., 2019)	lr_inst_param=0.2, wd_inst_param=0.0
SR (Zhou et al., 2021)	$\tau = 0.5, \lambda_0 = 0$, epochs=20
Co-teaching (Han et al., 2018b)	$T_k = 3, \tau = \epsilon$ (ϵ is the noise level)
CNLCU (Xia et al., 2022)	$T_k = 3, \tau_{min} = 0.1$, fixed-length time intervals=5
SEAL (Chen et al., 2021)	Number of iterations=4

Table 11: Optimal hyperparameter settings for different controlled synthetic label noise on NoisywikiHow.

Method	Noise Level				
	0%	10%	20%	40%	60%
	Top-1(Top-5)	Top-1(Top-5)	Top-1(Top-5)	Top-1(Top-5)	Top-1(Top-5)
BBN (Zhou et al., 2020)	63.11(**87.06**)	62.03(**86.79**)	60.03(**85.73**)	55.59(**83.68**)	50.22(80.47)
LDAM (Cao et al., 2019)	**64.25**(86.82)	**62.71**(86.19)	**60.69**(85.18)	**56.29**(82.53)	**50.79**(79.52)
Decoupling-NCM (Kang et al., 2020)	62.54(85.59)	60.85(85.61)	58.94(84.71)	54.86(82.58)	50.09(79.76)
Decoupling-cRT (Kang et al., 2020)	62.89(86.16)	61.86(86.53)	59.99(85.41)	55.80(83.29)	51.82(**81.40**)
Decoupling-LWS (Kang et al., 2020)	61.87(85.75)	60.42(85.96)	58.61(84.63)	54.30(82.20)	49.61(79.50)

Table 12: Top-1 (Top-5) test accuracy (%) of long-tailed learning methods on NoisywikiHow under different noise levels.

Method	**Noise Level**				
	0%	10%	20%	40%	60%
BBN (Zhou et al., 2020)	-				
LDAM (Cao et al., 2019)	C=0.2, s=10	C=0.5, s=10	C=0.7, s=7	C=0.8, s=7	C=0.8, s=10
Decoupling-NCM (Kang et al., 2020)	-				
Decoupling-cRT (Kang et al., 2020)	epoch'=5, num_samples_cls=4				
Decoupling-LWS (Kang et al., 2020)	epoch'=5, num_samples_cls=4				

Table 13: Optimal hyperparameter settings for different controlled real-world label noise on NoisywikiHow.

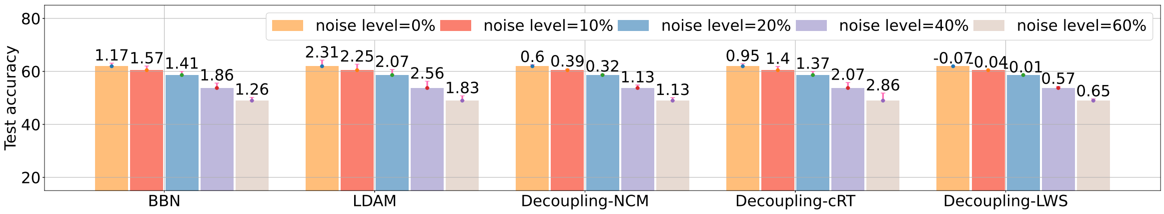


Figure 7: Performance improvements over the base model under different long-tailed learning methods on NoisywikiHow. Given a method (e.g., BBN) and a noise level, a column height reflects performance when only using the base model. The length of the pink line on the column represents the performance boost from adopting the method.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
See section Limitations.

□ A2. Did you discuss any potential risks of your work?
Not applicable. Left blank.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
See abstract and introduction.

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

See Section 3.

□ B1. Did you cite the creators of artifacts you used?
Not applicable. Left blank.

□ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Not applicable. Left blank.

□ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Not applicable. Left blank.

□ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Not applicable. Left blank.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
See Section 2.

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
See Section 4.1

C ☑ Did you run computational experiments?

See Section 4.

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
See Section 4.

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
See Section 4.

☒ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
We report results from another dimension, i.e., we simultaneously use Top-1 accuracy and Top-5 accuracy as the evaluation metrics.

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
See Appendix A.2.

D ☑ Did you use human annotators (e.g., crowdworkers) or research with human participants?

See Section 3.2 and Appendix A.3.

☑ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
See Appendix A.2 and A.3.

☑ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
See Section 3.

☑ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
See Appendix A.1.

□ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Not applicable. Left blank.

☑ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
See Section 3.