NOISEBENCH: Benchmarking the Impact of Real Label Noise on Named Entity Recognition

Elena Merdjanovska[1,2], Ansar Aynetdinov[1], Alan Akbik[1,2]
[1]Humboldt-Universität zu Berlin
[2]Science of Intelligence
{elena.merdjanovska, aynetdia, alan.akbik}@hu-berlin.de

Abstract

Available training data for named entity recognition (NER) often contains a significant percentage of incorrect labels for entity types and entity boundaries. Such *label noise* poses challenges for supervised learning and may significantly deteriorate model quality. To address this, prior work proposed various *noise-robust learning* approaches capable of learning from data with partially incorrect labels. These approaches are typically evaluated using *simulated noise* where the labels in a clean dataset are automatically corrupted. However, as we show in this paper, this leads to unrealistic noise that is far easier to handle than real noise caused by human error or semi-automatic annotation. To enable the study of the impact of various types of real noise, we introduce NOISEBENCH, an NER benchmark consisting of clean training data corrupted with 6 types of real noise, including expert errors, crowd-sourcing errors, automatic annotation errors and LLM errors. We present an analysis that shows that real noise is significantly more challenging than simulated noise, and show that current state-of-the-art models for noise-robust learning fall far short of their achievable upper bound. We release NOISEBENCH for both English and German to the research community[1].

1 Introduction

Named entity recognition (NER) is the task of detecting and classifying named entities in text, such as the names of organizations or locations. Current state-of-the-art approaches for NER still require supervision in the form of labeled training data (Zaratiana et al., 2023), i.e. sentences in which named entities are marked and assigned their correct type. However, prior work found that available datasets for NER and other supervised tasks are affected by *label noise*, meaning that a certain percentage of entity labels are incorrect. For instance, the common NER dataset CoNLL-03 (Tjong Kim Sang and De Meulder, 2003) was estimated in various studies to have noise shares of between 5 and 7% (Wang et al., 2019; Reiss et al., 2020; Rücker and Akbik, 2023). Other NER datasets have also been found to contain a share of incorrect labels, with OntoNotes4 estimated around 8% and WNUT-17 around 18% (Wang et al., 2019; Huang et al., 2021).

Label noise introduces inconsistencies during training, which may significantly deteriorate model quality (Zhang et al., 2021a). To address this issue, prior work proposed approaches for *noise-robust learning* aimed at mitigating the negative effects of the noisy training signal (Song et al., 2022). However, the evaluation of these approaches has two main limitations.

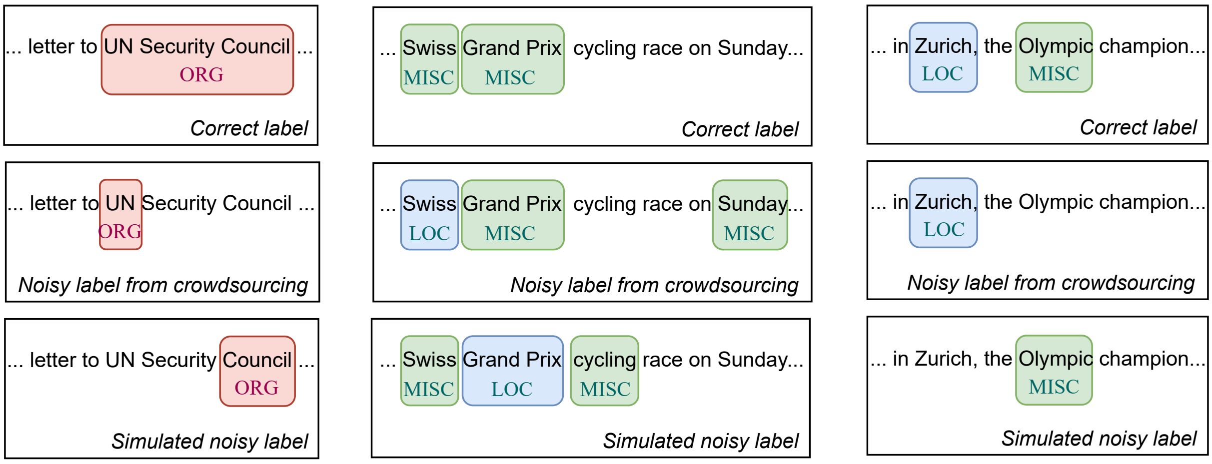
Limitation 1: Simulated noise is too easy. Most current research in noise-robust learning relies on experiments with *simulated label noise* (Tänzer et al., 2022; Klie et al., 2023). While this allows for evaluation in a controlled setting, it has been shown that simulated noise, even though it can model noise well to some extent, is much easier for deep learning models to disregard than real label noise (Jiang et al., 2020; Zhu et al., 2022).

Refer to Figure 1 for an illustrative comparison between real and simulated noise for three example sentences, including different types of errors that occur in NER datasets. These examples demonstrate that simulated noise can introduce similar errors as real noise, however the choice of spans to mislabel is random and as a result often less plausible. This means that an approach shown to be robust to simulated noise may not in fact be robust to real noise in practice.

Limitation 2: Distinct types of real noise. Additionally, there exist many possible sources of "real" noise. For instance, expert labelers may make different mistakes than crowd workers (Frenay and Verleysen, 2014). Next to human label-

[1]https://github.com/elenamer/NoiseBench

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 18182–18198
November 12-16, 2024



(a) Example of a **partial match** error induced by noise. Real noise makes a plausible mistake by labeling "UN" as ORG (organization), whereas simulated noise implausibly caused "Council" to be labeled.

(b) Examples of **type** and **non-entity** errors induced by noise. Real noise makes a plausible mistake by labeling "Swiss" as a LOC (location), whereas simulated noise implausibly labels "Grand Prix" as LOC. Real noise makes a plausible non-entity mistake by labeling "Sunday", whereas simulated noise labels "cycling".

(c) Example of a **missing** mention induced by noise. Real noise causes a plausible omission ("Olympic"), whereas simulated noise omits a trivial entity annotation ("Zurich")

Figure 1: Examples of text snippets with correct labels (top row) and two types of noise: Real noise from crowdsourcing (middle row) and simulated class-dependent noise (bottom row). This introduces different types of errors: (a) partial matches of correct entity mentions, (b) a wrong type and a non-entity mention and (c) a missing entity. We qualitatively find real noise to be more plausible than simulated noise.

ing, there are widely-used automatic approaches to create NER-labeled datasets such as distant supervision from a knowledge base (Mintz et al., 2009; Hedderich et al., 2021) and weak supervision using rules (Zhang et al., 2021b). Lastly, current research investigates the use of LLMs to label datasets (Golde et al., 2023; Wang et al., 2023).

We postulate that these types of real noise differ in their characteristics, meaning that a noise-robust learning approach shown to perform well on one type of noise may not perform well on another. For this reason, we argue there is a need for evaluating noise-robustness across multiple label noise types.

Contributions. With this paper, we present NOISEBENCH, a new benchmark for measuring the impact of label noise in the training data on the prediction quality of trained NER models. In more detail, our contributions are:

- We construct a noisy training dataset in 7 different variants, where each noisy variant contains the same sentences and is affected by one class of real errors, spanning errors made by experts, crowd workers, distant supervision, weak supervision and teacher LLMs.
- We present a set of experiments that empirically show that real noise from NOISEBENCH is significantly more difficult for current approaches. We further find that during training, real noise is memorized immediately, whereas memorization of simulated noise is delayed.
- We comparatively evaluate current state-of-the-art approaches for noise-robust learning on NOISEBENCH, and experimentally establish upper bounds.

Our analysis finds that no single current approach works best for all types of real noise, and that all current approaches fall far short of their theoretical upper bound. To enable the research community to leverage our benchmark in their evaluations, we publicly release all data and implementation.

2 NOISEBENCH

Our benchmark is derived from a subset of the classic CoNLL-03 dataset for NER in English, annotated with entities belonging to four classes. We chose this dataset since it has been extensively studied in the field, allowing us to integrate various prior works. We derive a similar benchmark for NER in German, in Section 5 and Appendix A.

NOISEBENCH consists of the following parts: (1) A noise-free test split to evaluate trained models. (2) Seven variants of the training split, where six are annotated with different types of noise and one is without noise. Table 1 presents the quality of the six noisy variants w.r.t. the noise-free dataset.

The training split contains 5,885 sentences from 400 documents, covering 9,685 entity mentions. The test split contains 3,427 sentences from 231 documents, covering 5,725 entity mentions.

2.1 Types of Noise

In the following, we discuss each training split and the type of noise it models.

2.1.1 Noise-Free Data

Our benchmark requires two splits without any label noise: A clean test split to evaluate models trained on noisy training data, and a **Clean** training split to measure the upper bound performance.

Since the original annotations of CoNLL-03 have been shown to be noisy (Wang et al., 2019; Reiss et al., 2020), we use the labels of CLEANCONLL (Rücker and Akbik, 2023), a recently released resource in which 7% of all original annotations were semi-automatically relabeled. In their evaluation, Rücker and Akbik (2023) find their resulting dataset to be of very high quality and largely improved consistency. The **Clean Test** split in our benchmark is the standard CoNLL-03 test split, with the CLEANCONLL labels.

2.1.2 Expert Errors

Most machine learning datasets are created using manual annotation by domain experts that provide high-quality labels. However, errors have been found to occur even in expert annotation, affecting even well-known benchmarks, though usually with relatively low noise shares of under 10% (Northcutt et al., 2021b; Song et al., 2022). To represent such noise, our benchmark includes a variant of the train split called **Expert**, which contains the original CoNLL-03 annotations. As Table 1 shows, this split has a noise share of 5.5% and is thus the split with lowest noise.

2.1.3 Crowdsourcing Errors

Crowdsourcing is a less costly alternative to expert annotation, but also more prone to annotation errors (Frenay and Verleysen, 2014). In order to create noisy variants of the train set representing real-world human errors, we utilize the crowdsourced labels by Rodrigues et al. (2014). This study involves 47 crowd workers labelling a subset of the English CoNLL-03 dataset, of around 400 news articles. They released their dataset and all annotations produced by each crowd worker. We selected only the sentences where the tokenization matched the Clean variant, resulting in 5,885 sentences.

We include two noisy training splits based on crowd annotations into our benchmark: (1) In the first, **Crowd**, we do a simple majority vote over all annotations provided for each token, i.e. the baseline method for aggregating crowdsourced annotations. (2) In the second, **Crowd++**, we use an oracle version of the majority vote, selected by either taking the correct label if it is provided by any of the annotators or, in the absence of a correct label, by choosing the label with the majority of votes. This version represents the upper bound of crowdsourced labels given a perfect label aggregation method. As Table 1 shows, the noise share of **Crowd** (36.6%) is considerably higher than **Crowd++** (15.3%).

2.1.4 Distant Supervision

One approach for labeling data without human participation is *distant supervision* (Mintz et al., 2009), where entity mentions in target datasets are matched to entity types in knowledge bases (KBs).

We include a **Distant** noisy training variant in our benchmark, adapted from the annotations by Liang et al. (2020)[2] that use the Wikidata corpus and gazetteers collected from multiple online sources as external knowledge bases. After initial POS tagging, the unlabeled sentences were matched with the knowledge bases. This process results in incomplete annotations due to limited coverage over entity types of KBs. This explains the rather high number of missing entities and the overall noise level (31.3%) of the **Distant** training variant, as shown in Table 1.

2.1.5 Weak Supervision

Another approach aimed at reducing manual annotation efforts is weak supervision. Here, labels are obtained using a number of "weak" supervision sources, such as heuristics or expression-based rules. Each weak source is typically specialized to detect only a subset of the correct labels.

We use the labels from the approach by Lison et al. (2020)[2] to create our **Weak** label set. This covers 16 weak labeling sources (Zhang et al., 2021b), including heuristics, gazetteers and predictions of NER models trained on other corpora. An example heuristic is detecting PER (person) entities using a pre-defined list of first names.

We aggregate the weak label sets with simple majority voting. We apply majority vote on every token with at least one entity label assigned to it,

[2] Available under Apache 2.0 license

Noisy train split	*%Noise*	*F1*token	*F1*entity	*#Entities* Total	*#Entities* Correct	*%Errors* Missing (FN)	*%Errors* Non-entity (FP)	*%Errors* Type	*%Errors* Partial
Expert	5.5	99.0	94.5	9,644	9,129	10.0	2.8	74.0	13.3
Crowd++	15.3	96.7	84.7	8,607	7,751	59.6	8.7	17.0	14.7
Crowd	36.6	92.3	63.4	7,188	5,352	61.9	10.2	16.0	11.9
Distant	31.3	92.9	68.7	7,329	5,846	65.4	10.5	12.9	11.1
Weak	40.4	91.9	59.6	10,640	6,058	17.4	34.6	36.3	11.8
LLM	45.6	87.4	54.4	11,349	5,726	22.5	45.4	28.3	3.7

Table 1: Overview of the noisy training splits in NOISEBENCH. The table shows the noise level, the micro-averaged token-level F1 score (*F1*token), micro-averaged entity-level F1 (*F1*entity), the number of entities (*Total*), number of correct entities (*Correct*) and share of each error type: missing mentions (*Missing (FN)*), non-entity mentions (*Non-entity (FP)*), wrong type (*Type*) and partial matches (*Partial*). All metrics are in reference to the Clean split.

following Zhang et al. (2021b). Due to the large number of labelling sources, majority voting yields a large number of entities, as shown in Table 1, including many false positives. As a result, the **Weak** label set has a high noise share of 40.4%.

2.1.6 LLM Teacher Models

Our benchmark includes a noisy variant of the train split annotated by an LLM. This follows recent efforts that use LLMs for dataset generation (Wang et al., 2023). Here, the main idea is to pass a description of the annotation task and target classes to an LLM, and provide sentences to label. LLMs are able to generate high quality labels for some tasks (e.g. sentiment classification) while for others (e.g. NER and question type categorization) the resulting labels are very noisy (Golde et al., 2023).

We created the **LLM** variant using the Fabricator toolkit (Golde et al., 2023) by prompting GPT3.5 for named entities in our training dataset. To use LLM outputs for annotation of NER datasets, a certain output format is required. To achieve this, we provide one example with the correct output format in each prompt. This example is the same for each sentence we wish to annotate, which we refer to as a static one-shot setting. The example sentence was selected from the remainder of the CoNLL-03 training split, which consists of all sentences not included in our benchmark.

As Table 1 shows, the LLM label set results in the highest noise share of 45.6%. This is mainly due to the large number of nouns incorrectly identified as entity mentions, which also makes this the label set with the largest number of entity annotations out of the variants in NOISEBENCH.

2.2 Statistics

An overview of NOISEBENCH is given in Table 1. The table shows token-level F1 score and entity-level F1 score expressed as percentages. We define the noise level (*%Noise*) in terms of the entity-level F1 score, as $100 - \%F1$. The noise levels of the noisy splits range from 5.5 to 45.6 percent.

The table also shows the share of different error types. The errors are categorized into 4 main categories: **missing** mentions, **non-entity** mentions (false positives), incorrect entity **type** (where the boundary is correct, but type incorrect) and **partial** matches. Partial matches are special cases where the type is correct, but the mention boundary is only partially correct. Refer to Figure 1 for examples.

We observe that the Crowd++, Crowd and Distant label sets have a lower total number of entity annotations than the Clean dataset, and the largest portion of errors are missing mentions. Conversely, the Weak and LLM label sets have more annotations than the Clean dataset, and most of the errors are either an incorrect mention or incorrect type. Most of the errors in the Expert label set are due to incorrect type. Regarding the number of partial matches, for almost all noise types, they make up between 10% and 15% of all errors.

3 Comparing Real and Simulated Noise

We first use NOISEBENCH to investigate how real label noise affects NER model performance in comparison to simulated noise. For this, we conduct two experiments: the first one addresses the impact of each type of training noise on the clean test set performance, and the second one compares training dynamics under real and simulated label noise to highlight the differences in noise memorization.

3.1 Noise Simulation Methods

We consider two noise simulation methods, namely the simple *uniform noise* used in most prior work and a more involved *oracle class-dependent noise* method that we design to mirror each noisy variant in NOISEBENCH.

Uniform noise. Uniform noise corrupts samples into any other label with a uniform probability distribution, given a target noise share. Studies investigating simulated noise in the NER task commonly rely on variants of this method (Mayhew et al., 2019; Tänzer et al., 2022).

Oracle class-dependent noise. Class-dependent noise is based on the knowledge that some pairs of classes are more likely to be mislabeled than others. It is defined by a noise transition matrix, which contains the mislabeling probabilities between all pairs of classes (Hedderich et al., 2021). We design an oracle version of class-dependent noise, where the per-class mislabeling probabilities of real noise are known. This allows us to investigate class-dependent noise in an ideal case, where it is able to mirror real noise closely, even though this is not possible in practice. This method mirrors real noise by utilizing the token-level mislabeling frequencies as probabilities to form a noise transition matrix.

Using each noise simulation method, we created 6 label sets, corresponding to each noise level in NOISEBENCH. It should be noted that the simulated labels replicate the token-level F1 scores of the real noisy labels, however the entity-level F1 and sentence-level accuracy can deviate.

3.2 Experimental Setup

In both experiments, we train a baseline approach for NER on each noisy variant of the training split, as well as on the additional simulated noise.

Validation splits. We evaluate the setting in which all available data to train a model is noisy, including the validation set. To obtain noisy validation sets for each of our 7 dataset variants, we split the noisy datasets into training and validation sets. All sentences from 66 news documents from 1996-08-24 comprise the validation set, which is around 17% of all sentences, and are left out from model training and used for hyperparameter tuning.

Baseline. For NER, as a baseline approach, we fine-tune an `xlm-roberta-large` transformer using the FLERT approach (Schweter and Akbik, 2021). It improves upon the regular fine-tuning setup by considering document-level features of a sentence to be tagged. We use a learning rate of 5e-6 and a batch size of 32, for a fixed number of 10 epochs, without early stopping. These parameters were obtained according to the performance on a noisy validation set, keeping in mind that larger batch sizes are more robust to noise (Rolnick et al., 2017). We use entity-level micro F1-score.

3.3 Experiment 1: Impact of Label Noise on Test Performance

In the first experiment, we compare how the clean test set performance is impacted by the 6 types of real label noise when present in the training set. In addition, we provide the same comparison for corresponding simulated noisy label sets.

3.3.1 Results and Discussion

The results for uniform noise are shown in Appendix D. We initially established that uniform noise is less challenging for the model, so in the results for Experiments 1 and 2 we chose to focus solely on oracle class-dependent noise.

The main results from Experiment 1 for oracle class-dependent noise are shown in Table 2. Following are our main observations.

Label noise degrades performance. When we compare the test F1 scores of the real noisy variants with the average score of 93.99 achieved when training on the Clean variant, we see that model performance is affected by each noise type. As the noise levels increase, this impact becomes more pronounced, showing that the baseline model lacks robustness to real noise. Comparing the test F1 scores of the simulated noisy variants, we can see that noise of 5.9% in the training set results in a score comparable to training on the Clean variant. However, as simulated noise levels increase, the noise does degrade test set scores.

Real noise is more difficult. Furthermore, when we compare the real noisy label sets with their equivalent simulated noisy variants, we can observe that the simulated training variants show a score of around 2.5 percentage points higher on average than the real label sets. This shows that

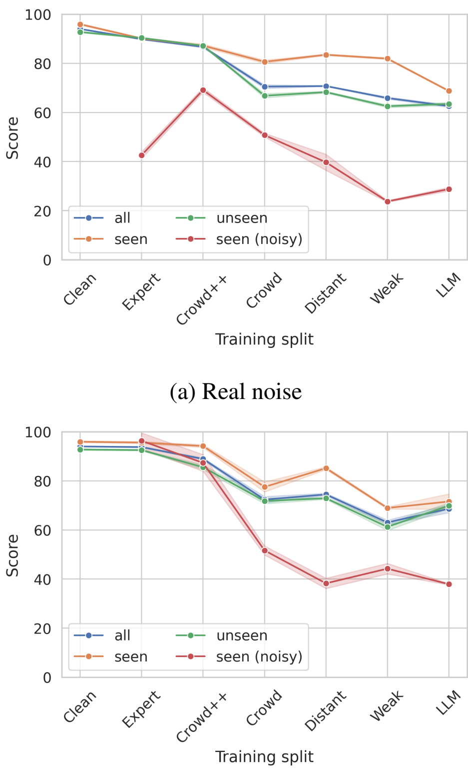
	Real noise		Simulated noise		Δ
	%Noise	*F1*	*%Noise*	*F1*	*F1*
Clean	0	94.0 ±0.0	-	-	-
Expert	5.5	89.8±0.2	5.9	93.7±0.2	3.9
Crowd++	15.3	86.7±0.3	17.9	88.9±0.4	2.2
Crowd	36.6	70.5±0.6	41.3	72.4±1.0	1.8
Distant	31.3	70.8±0.1	39.2	74.5±0.4	3.7
Weak	40.4	65.9±0.4	41.2	63.1±0.8	-2.8
LLM	45.6	62.6±0.4	47.2	68.6±1.3	6.0
Average		74.4±0.3		76.9±0.7	2.5

Table 2: F1 scores on the Clean Test split of the baseline FLERT approach, fine-tuned on different noisy variants of the training set. The scores are averages of 3 runs. The column Δ (difference) refers to the difference in F1 score on the test split when training on a dataset with real noise compared to simulated class-dependent noise.

for predictive NER models, real noise is more difficult to overcome than simulated noise. In other words, models are more likely to overfit to real noisy labels, rather than simulated ones.

Models generalize to unseen entities well. Figure 2 shows F1 scores for seen and unseen entities separately, further distinguishing seen entities by whether their label in the training set was clean or noise-corrupted. In Figure 2a we see that for the Expert and Crowd++ noise types, the score on the seen (clean) and the unseen entities is comparable, which indicates the model has the ability to generalize to unseen entities well. As for the remaining training splits with noise levels of over 30%, noise also affects the performance on unseen entities.

Simulated noisy patterns are disregarded. For all real noise types, the score on the seen (noisy) entities is low. With simulated noise however, in Figure 2b we see that for Expert and Crowd++, the score on the seen-noisy entities and seen-clean entities is close. This means that at low noise levels, the models are able to disregard simulated noisy patterns and predict the same entities correctly when they appear in the test set.



(a) Real noise

(b) Simulated class-dependent noise

Figure 2: F1 scores on different subsets of entities in the test set: all, seen (clean), seen (noisy) and unseen.

3.4 Experiment 2: Memorization of Noise

Prior analysis has found that there are distinct phases of learning when training a model on data with label noise (Arpit et al., 2017). This has been referred to as a *generalization phase*, where models learn patterns that generalize well to clean data, followed by a *memorization phase*, where models overfit to the label noise and deteriorate in prediction quality (Tänzer et al., 2022).

To investigate this phenomenon for real and simulated noise, we extend the training stage to 100 epochs. At the end of each epoch, we measure the F1 score of the model on both the noisy training split it is being trained on, and separately on the clean training split. The difference between these two scores allows us to measure memorization.

3.4.1 Results and Discussion

In Figure 3 we show training curves from training with real and simulated variants of NoiseBench for 3 noise types: Expert, Crowd++ and Distant. We plot two scores: the F1-score on the respective noisy variant of the training set, and the F1 score on the Clean variant of the training set. In all training curves, we can observe the memorization effect, with each model perfectly fitting the noisy data by the end of training and reaching an F1 score close to 1.

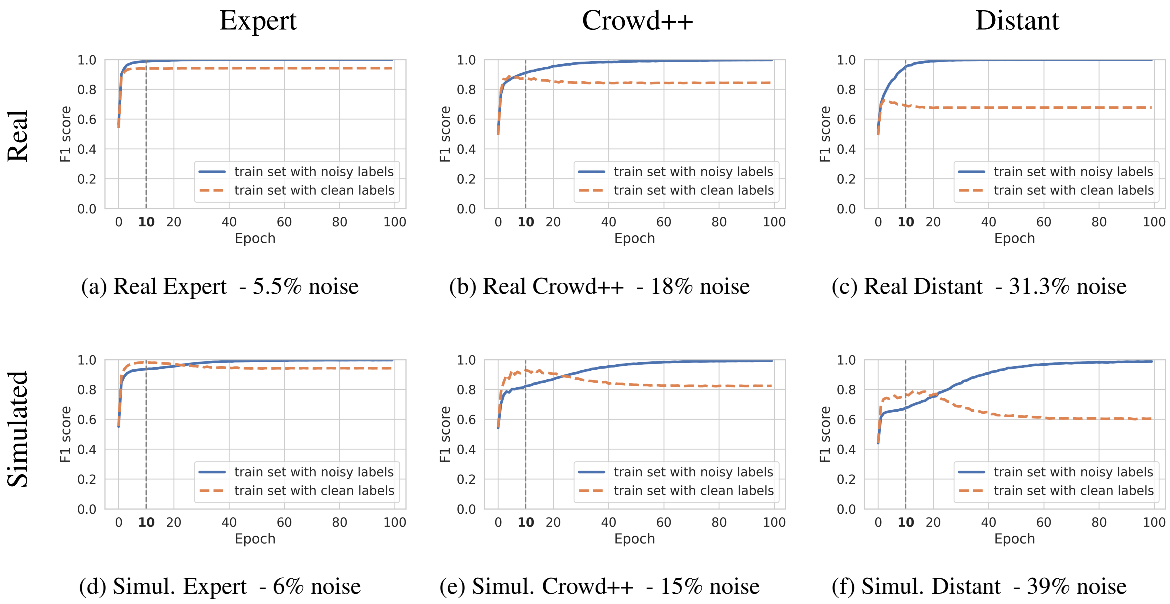
Delayed memorization of simulated noise. However, we note that with simulated noise (see Figure 3d, 3e, 3f) this happens much later in the training process than with real noise. In addition, the training curves of simulated noise show a stage during the early epochs where the score on the clean labels is consistently higher than the score on the noisy labels. This confirms previous findings that the model is able to learn general patterns first, before starting to memorize the noise.

Immediate memorization of real noise. With real noise, this does not happen and the model starts fitting the noisy labels from the beginning (see Figure 3a, 3b, 3c). As a result, the score on the clean labels is consistently lower than the score on the noisy labels, throughout the training run[3].

Our experiments find that real noise does not display distinct generalization/memorization phases during training, and rather immediately begins with memorization[4]. This makes intuitive sense, as real noise has underlying patterns that may be extracted

[3]We confirm this finding for German in Appendix A.2.2.

[4]We confirm this finding for a smaller model, as well as a randomly initialized model in Appendix E.



(a) Real Expert - 5.5% noise (b) Real Crowd++ - 18% noise (c) Real Distant - 31.3% noise

(d) Simul. Expert - 6% noise (e) Simul. Crowd++ - 15% noise (f) Simul. Distant - 39% noise

Figure 3: Comparison of model performance during extended training. The top row shows models fine-tuned on label sets with real noise, while the bottom row models fine-tuned on corresponding simulated (class-dependent) noisy labels. The plots are averages of 3 runs. The graphs for Crowd, Weak and LLM are shown in Appendix E.

during learning. This lends further evidence to the increased challenges and the need to evaluate noise-robust learning with real noise.

4 Evaluating Noise-Robust Learning

Having established the difficulty of real noise, we now use NOISEBENCH to perform a comparative evaluation of widely-used noise-robust learning approaches. Our goal is to determine their effectiveness in the presence of real label noise, and to establish upper bounds of what noise-robust learning could ideally achieve.

4.1 Compared Approaches

We surveyed current state-of-the-art methods for noise-robust NER and found that many approaches rely on the same underlying ideas for handling label noise. In the following, we group approaches by the underlying idea, select a state-of-the-art representative for each group and, if possible, derive an upper bound method for each group. For more details about the implementation of compared approaches refer to Appendix B.

4.1.1 Learning from a Clean Subset

The first family of approaches relies on utilizing the subset of each noisy dataset in which all labels are correct. One type of these approaches filters out all likely incorrect annotations and learns only from a clean subset. Another type derives confidence weights for each sample so that annotations judged to be of higher quality feature more during training.

As a representative of the former type of approaches targeting clean subsets of noisy datasets, we chose Confident Learning (Northcutt et al., 2021a), while the latter type is represented by CrossWeigh (Wang et al., 2019) and Learn-To-Reweight (L2R) (Ren et al., 2018).

Upper bound: Oracle subset. To obtain an upper bound for this family of approaches, we use an oracle to select the subset of clean sentences from each of the noisy training splits in NOISEBENCH. We then use the baseline fine-tuning approach only on this subset, illustrating a best-case scenario.

4.1.2 Delaying Memorization

Another family of noise-robust learning approaches seeks to leverage the two phases of learning (generalization and memorization) we discussed in Section 3.4. They seek to either draw our the generalization phase or cease training before memorization begins. While our experiments indicate that these two phases do not exist for real noise, we nevertheless include this family of approaches in our evaluation since they are widely used. As representative of this class of approaches, we chose co-regularization (Zhou and Chen, 2021).

Upper bound: Oracle stopping. To obtain an upper bound for this family of approaches, we use a simple stopping criterion based on the score on the clean test set at the end of each epoch. We use the epoch of best generalization to report the final score. This simulates an ideal stopping.

4.1.3 Combined Approaches

While the approaches discussed so far each build on the individual ideas of identifying a clean subset or delaying memorization, many current approaches in fact combine multiple of such ideas in multi-stage pipelines (Liang et al., 2020; Yu et al., 2021; Wang et al., 2022). As representative of such approaches, we evaluate BOND (Liang et al., 2020) and meta self-refinement (MSR) (Zhu et al., 2023a), both of which combine pseudo-labeling in a student-teacher setup and confidence-based sample selection.

No upper bound for pseudo-labeling. We cannot derive a separate upper bound for pseudo-labeling, as the best case scenario here would mean that all noisy labels are replaced by correct labels, which is the same as training on the Clean dataset.

4.1.4 Additional Clean Data

We include a further upper bound for the scenario in which a small amount of high quality noise-free data is available. This is inspired by the extensive analysis of the use of clean validation data in Zhu et al. (2023b). Here, after first training on the noisy training set, they use a small clean dataset to continue fine-tuning the model. We include this upper bound to measure the accuracy gains that may be achieved if one were to invest effort in manually annotating additional noise-free data.

4.2 Results

Table 3 summarizes the evaluation results. We make the following observations:

Identifying a clean subset has highest potential. The upper bound of training only the clean subset of each noisy split (see "Oracle subset" in Table 3) achieves the best scores of all upper bounds. This makes intuitive sense as training is performed only over fully clean sentences, albeit a smaller subset of the full training data as all noisy sentences are filtered out. Similarly, we find strong improvements for the "Additional clean data" upper bound. Oracle stopping, on the other hand, does not achieve the same level of performance as the oracle subset, only slightly outperforming the FLERT baseline. This is in line with our findings in Experiment 2 that the early-learning generalization phase is skipped when training with real noise. This indicates that noise-robust learning approaches that target early stopping have little potential.

Small benefit of noise-robust approaches. Evidently, there is no single best approach for all noise types. For each noise type, at least one noise-robust approach outperforms the baseline, however on average most of them are comparable to it. Only MSR outperforms the baseline averaged over all noise types, bringing improvements for Crowd++, Distant, Weak and LLM. Additionally, L2R works well for LLM noise and BOND for Crowd. Still, the performance is far below the upper bound. This raises the issue of trade-offs of existing noise-robust learning approaches, since they often require additional hyperparameter tuning or incur computational costs, but only lead to slight improvements in the presence of real noise.

5 Ablation: NoiseBench for German

Using the German sentences in CoNLL-03, we created a noisy label benchmark for German to confirm our findings for a different language. Following the English counterpart described in Section 2, it consists of (1) a noise-free test split to evaluate trained models and (2) three variants of the training split, where two are annotated with different types of noise and one is the ground truth. The two types of noise include Expert labels, with 16.2% noise and LLM labels, with 54% noise. More details can be found in Appendix A.1.

5.1 Experimental Results

Table 4 shows the results of the noise-robust approaches and upper bounds when training on the German datasets. More experimental details and results can be found in Appendix A.2.

Oracle subset score reaches an upper limit. Regarding the upper bounds, we see that the performance of the oracle subset of Expert and LLM is close, meaning that the 4000[5] clean sentences in the LLM subset are already enough to reach an F1 score over 82. Despite having more samples, the Expert subset does not result in a much higher score. This could signify that the remaining sentences, not included in the Expert subset, are difficult examples necessary to properly learn the task.

Poor performance of noise-robust approaches. Regarding the noise-robust approaches, only Confident Learning is able to match and slightly outperform the baseline. All other methods mainly perform poorly on the German dataset, even below

[5] See Appendix C for the size of the oracle subset.

	Clean	Expert	Crowd++	Crowd	Distant	Weak	LLM	Avg.
Baseline	93.99±0.04	89.84±0.19	86.71±0.29	70.52±0.62	70.75±0.13	65.87±0.36	62.60±0.39	77.18
Upper bounds								
Oracle subset	-	**90.31**±0.28	**91.83**±0.33	**85.95**±0.59	**83.07**±0.59	**81.13**±1.25	**75.70**±1.10	**85.99**
Oracle stopping	94.06±0.07	89.88±0.19	87.23±0.24	71.04±0.88	71.84±0.61	66.98±0.22	63.64±0.44	77.81
Additional clean data	**94.14**±0.18	90.04±0.30	89.14±0.67	81.70±0.95	80.19±0.73	71.66±1.66	72.06±1.32	82.70
Noise-robust learning								
Confident learning	**93.71**±0.31	**90.01**±0.15	86.53±0.23	69.99±0.97	71.41±0.34	65.81±0.46	61.75±0.56	77.03
CrossWeigh	93.50±0.12	89.68±0.49	85.01±0.83	64.95±1.18	70.55±0.24	66.15±0.24	60.87±1.40	75.82
L2R	90.29±2.12	82.10±4.10	79.91±2.27	67.51±1.01	65.45±2.01	63.36±0.34	**65.29**±4.15	73.42
Co-regularization	93.65±0.11	89.55±0.22	86.91±0.31	72.22±0.73	70.45±0.22	65.52±0.62	62.23±0.76	77.22
BOND	89.92±0.71	86.78±0.35	86.13±0.81	**74.12**±0.49	73.62±0.70	66.60±0.36	60.99±0.77	76.88
MSR	92.83±0.16	89.53±0.48	**88.45**±1.08	68.44±3.79	**75.80**±1.41	**69.48**±0.32	64.57±1.22	**78.44**

Table 3: Performance of noise-robust approaches on the Clean test set, when training on NOISEBENCH training split variants. Results are expressed in terms of F1 score. Each score is averaged over 3 runs.

	Clean	Expert	LLM	Avg.
Baseline	90.24±0.2	79.02±0.4	57.86±0.4	75.7
Upper bounds				
Oracle subset	-	**83.11**±0.6	**82.72**±0.6	82.9
Oracle stopping	90.50±0.2	79.48±0.3	61.81±0.9	77.3
Additional clean data	89.86±0.9	82.85±1.7	69.50±1.4	80.7
Noise-robust learning				
Confident learning	90.00±0.3	**79.57**±0.3	58.03±0.2	75.9
CrossWeigh	**90.11±0.3**	78.32±0.3	57.50±0.8	75.3
L2R	81.45±3.1	74.14±0.8	53.07±1.6	69.6
Co-regularization	88.50±0.2	78.49±0.2	54.47±0.5	73.8
BOND	86.53±0.3	77.56±0.5	55.89±0.6	73.3
MSR	85.34±0.5	76.42±0.4	**64.00**±0.7	75.3

Table 4: German variant: Performance of noise-robust approaches on the Clean test set, when training on each training split variant. Results are expressed in terms of F1 score. Each score is averaged over 3 runs.

the baseline, with the exception of the improvement brought by MSR on the LLM dataset.

6 Related Work

There are a few benchmarks for learning with label noise and related areas. The WRENCH benchmark (Zhang et al., 2021b) focuses only on weak supervision labels for multiple tasks, with the emphasis on combining multiple weak labelling sources. Klie et al. (2023) compare a large number of methods for the detection of annotation errors. Multiple tasks are included, including NER on CoNLL-03, where they evaluate the detection of expert errors, concluding that most approaches are not successful at this. Similarly, Chong et al. (2022) evaluate annotation error detection on datasets with noise only from crowdsourced labels, for part-of-speech tagging and natural language inference tasks. Liu et al. (2022) propose a benchmark for text classification under label noise, where they re-annotate an existing sentiment classification dataset and construct noisy label sets according to annotator disagreements; however, they do not publish these label sets. NoisyWikiHow, a benchmark for intention identification has also been presented (Wu et al., 2023), where the authors propose a method to simulate realistic noise that imitates human errors by producing heterogeneous and instance-dependent errors. For NER in Estonian, Hedderich et al. (2021) introduce the NoisyNER, which includes multiple noise levels obtained from distant supervision approaches with varying quality. MultiCoNERv2 (Fetahu et al., 2023) addresses textual noise in the input data itself (e.g. typos), instead of label noise.

7 Conclusion

In this paper, we address the issue of label noise in the NER task. We introduce a new benchmark, based on the commonly used NER dataset CoNLL-03, for evaluating the impact of 6 distinct types of real label noise on the same set of sentences, with varying degrees of difficulty.

We demonstrated that real noise causes transformer-based language models to immediately memorize the noise pattern, making real label noise a more challenging problem than simulated label noise, even in the case of oracle class-dependent noise informed by the characteristics of real noise.

We further presented an evaluation of popular noise-robust learning approaches. Our experiments indicate that current methods fall far short of what can potentially be achieved on the noise types in NOISEBENCH and that approaches that focus on automatically identifying a clean subset of labels have the highest potential. We hope that NOISEBENCH aids other researchers in the further development of more effective noise-robust approaches.

Limitations

This paper focuses on the scenario when the entire available dataset could be noisy and we do not have access to a small, high-quality labelled, data subset. While this is a certainly scenario which reflects a large number of real-world cases, it could be argued that in some situations it is realistic to have the resources to ensure a subset of the data is clean, with high-quality annotations. However, when this is the case, Zhu et al. (2023b) showcased that this clean data would be better utilized by directly fine-tuning the models on it, instead of using it for validation. Therefore, we argue that this alternative setup is not particularly useful for the evaluation of label-noise-robust approaches.

Acknowledgements

We thank all reviewers for their valuable comments. Elena Merdjanovska and Alan Akbik are supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC 2002/1 "Science of Intelligence" – project number 390523135. Ansar Aynetdinov and Alan Akbik are supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Emmy Noether grant "Eidetic Representations of Natural Language" (project number 448414230).

References

Devansh Arpit, Stanisław Jastrzębski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, et al. 2017. A closer look at memorization in deep networks. In *International conference on machine learning*, pages 233–242. PMLR.

Derek Chong, Jenny Hong, and Christopher Manning. 2022. Detecting label errors by using pre-trained language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9074–9091, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Besnik Fetahu, Zhiyu Chen, Sudipta Kar, Oleg Rokhlenko, and Shervin Malmasi. 2023. MultiCoNER v2: a large multilingual dataset for fine-grained and noisy named entity recognition. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 2027–2051, Singapore. 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.

Jonas Golde, Patrick Haller, Felix Hamborg, Julian Risch, and Alan Akbik. 2023. Fabricator: An Open Source Toolkit for Generating Labeled Training Data with Teacher LLMs. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Association for Computational Linguistics.

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.

Xiusheng Huang, Yubo Chen, Shun Wu, Jun Zhao, Yuantao Xie, and Weijian Sun. 2021. Named entity recognition via noise aware training mechanism with data filter. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4791–4803, Online. Association for Computational Linguistics.

Lu Jiang, Di Huang, Mason Liu, and Weilong Yang. 2020. Beyond synthetic noise: Deep learning on controlled noisy labels. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 4804–4815. PMLR.

Jan-Christoph Klie, Bonnie Webber, and Iryna Gurevych. 2023. Annotation Error Detection: Analyzing the Past and Present for a More Coherent Future. *Computational Linguistics*, 49(1):157–198.

Chen Liang, Yue Yu, Haoming Jiang, Siawpeng Er, Ruijia Wang, Tuo Zhao, and Chao Zhang. 2020. Bond: Bert-assisted open-domain named entity recognition with distant supervision. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Pierre Lison, Jeremy Barnes, Aliaksandr Hubin, and Samia Touileb. 2020. Named entity recognition without labelled data: A weak supervision approach. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1518–1533, Online. Association for Computational Linguistics.

Bo Liu, Wandi Xu, Yuejia Xiang, Xiaojun Wu, Lejian He, Bowen Zhang, and Li Zhu. 2022. Noise learning for text classification: A benchmark. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 4557–4567, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Stephen Mayhew, Snigdha Chaturvedi, Chen-Tse Tsai, and Dan Roth. 2019. Named entity recognition with partially annotated training data. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 645–655, Hong Kong, China. Association for Computational Linguistics.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. 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 1003–1011.

Curtis Northcutt, Lu Jiang, and Isaac Chuang. 2021a. Confident learning: Estimating uncertainty in dataset labels. *J. Artif. Int. Res.*, 70:1373–1411.

Curtis G Northcutt, Anish Athalye, and Jonas Mueller. 2021b. 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)*.

Frederick Reiss, Hong Xu, Bryan Cutler, Karthik Muthuraman, and Zachary Eichenberger. 2020. Identifying incorrect labels in the conll-2003 corpus. In *Conference on Computational Natural Language Learning*.

Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. 2018. Learning to Reweight Examples for Robust Deep Learning. In *ICML*.

Filipe Rodrigues, Francisco Pereira, and Bernardete Ribeiro. 2014. Sequence labeling with multiple annotators. *Machine learning*, 95:165–181.

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

Susanna Rücker and Alan Akbik. 2023. CleanCoNLL: A Nearly Noise-Free Named Entity Recognition Dataset. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*.

Stefan Schweter and Alan Akbik. 2021. FLERT: Document-Level Features for Named Entity Recognition.

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*.

Michael Tänzer, Sebastian Ruder, and Marek Rei. 2022. Memorisation versus generalisation in pre-trained language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7564–7578, Dublin, Ireland. Association for Computational Linguistics.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Haobo Wang, Ruixuan Xiao, Yiwen Dong, Lei Feng, and Junbo Zhao. 2022. ProMix: combating label noise via maximizing clean sample utility. *arXiv preprint arXiv:2207.10276*.

Xiao Wang, Wei Zhou, Can Zu, Han Xia, Tianze Chen, Yuan Zhang, Rui Zheng, Junjie Ye, Qi Zhang, Tao Gui, Jihua Kang, J. Yang, Siyuan Li, and Chunsai Du. 2023. Instructuie: Multi-task instruction tuning for unified information extraction. *ArXiv*, abs/2304.08085.

Zihan Wang, Jingbo Shang, Liyuan Liu, Lihao Lu, Jiacheng Liu, and Jiawei Han. 2019. Crossweigh: Training named entity tagger from imperfect annotations. In *Conference on Empirical Methods in Natural Language Processing*.

Tingting Wu, Xiao Ding, Minji Tang, Hao Zhang, Bing Qin, and Ting Liu. 2023. NoisywikiHow: A benchmark for learning with real-world noisy labels in natural language processing. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4856–4873, Toronto, Canada. Association for Computational Linguistics.

Yue Yu, Simiao Zuo, Haoming Jiang, Wendi Ren, Tuo Zhao, and Chao Zhang. 2021. Fine-tuning pre-trained language model with weak supervision: A contrastive-regularized self-training approach. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1063–1077.

Urchade Zaratiana, Nadi Tomeh, Pierre Holat, and Thierry Charnois. 2023. GLiNER: Generalist Model for Named Entity Recognition using Bidirectional Transformer.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. 2021a. Understanding deep learning (still) requires rethinking generalization. *Commun. ACM*, 64(3):107–115.

Jieyu Zhang, Yue Yu, Yinghao Li, Yujing Wang, Yaming Yang, Mao Yang, and Alexander Ratner. 2021b. WRENCH: A comprehensive benchmark for weak supervision. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Wenxuan Zhou and Muhao Chen. 2021. Learning from noisy labels for entity-centric information extraction. *arXiv preprint arXiv:2104.08656*.

D. 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*, abs/2204.09371.

Dawei Zhu, Xiaoyu Shen, Michael Hedderich, and Dietrich Klakow. 2023a. Meta Self-Refinement for

Robust Learning with Weak Supervision. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1043–1058, Dubrovnik, Croatia. Association for Computational Linguistics.

Dawei Zhu, Xiaoyu Shen, Marius Mosbach, Andreas Stephan, and Dietrich Klakow. 2023b. Weaker than you think: A critical look at weakly supervised learning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14229–14253, Toronto, Canada. Association for Computational Linguistics.

A NoiseBench for German: Additional Details

Following are additional details about the German version of NoiseBench, as well as results from Experiment 2 for this dataset.

A.1 Overview

The training split contains 12,705 sentences from 553 documents, covering 10,008 entity mentions. The test split contains 3,160 sentences from 155 documents, covering 3,051 entity mentions.

A.1.1 Noise-Free Data

As noise-free labels for the German part of CoNLL-03, we take the updated annotations from 2006[6]. This updated label set is considered ground-truth by the research community.

When compared to the original CoNLL-03 German labels, he most changes in this label set are for the MISC class, most notably the removal of adjectives derived from names. This can alternatively be considered an update in annotation guidelines. With this, it should be noted that for the German dataset we do not have access to labels with verified high quality, as we do for the English counterpart with the CleanCoNLL labels.

A.1.2 Expert Errors

Similar as for English, we take the original CoNLL-03 labels (Tjong Kim Sang and De Meulder, 2003) as labels with expert errors. This results in a noise share of 16.2%.

A.1.3 LLM Teacher Models

Similar as for English, we use GPT3.5 to create a noisy version of the training split annotated by an LLM. This results in a high noise share of 54%.

[6]More details about the revision of the labels can be found in the ner.tz file, downloaded from `https://www.clips.uantwerpen.be/conll2003/ner/`, more specifically in the /ner/etc.2006/revision.txt and ner/etc.2006/guide.pdf files.

A.1.4 Statistics

When compared to the noise shares for Expert and LLM in NoiseBench in Table 1 (5.5% and 45.6%), the noise shares for German are higher (16.2% and 54% respectively). This is due to LLMs performing more poorly on languages other than English, as well as due to fewer research efforts focusing on re-annotating and cleaning the German part of CoNLL-03, resulting in less consistent labels.

We see that most of the errors in German are non-entity mentions. Overall, we also note that type errors and partial matches are much less prominent here (less than 10%), even though they formed a larger part of the errors in NoiseBench.

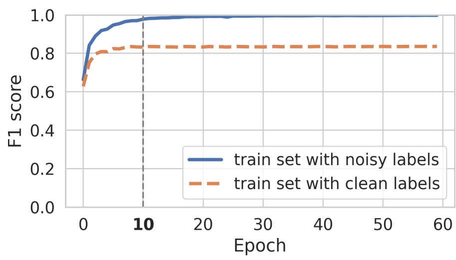
A.2 Experiments

A.2.1 Validation Split

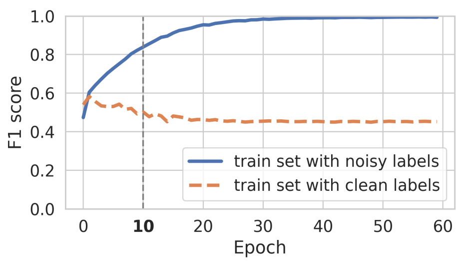
We take the last 96 documents from the training split to serve as a validation set, corresponding to roughly 17% of all sentences.

A.2.2 Experiment 2

We performed Experiment 2 for the German version of NoiseBench, where the goal is to observe the memorization of label noise. The resulting graphs for both noise types are shown in Figure 4. We observe a similar behaviour as in the English part of NoiseBench, where the real noisy datasets are memorized immediately.



(a) Expert - 16.4% noise



(b) LLM - 54.7% noise

Figure 4: Comparison of model performance during extended training, for the German dataset.

				#Entities		%Errors			
Noisy train split	*%Noise*	$F1_{\text{token}}$	$F1_{\text{entity}}$	*Total*	*Correct*	*Missing (FN)*	*Non-entity (FP)*	*Type*	*Partial*
Expert	16.2	98.2	83.8	11852	9156	14.5	73.0	6.1	6.4
LLM	54.0	92.1	46.0	16526	6102	19.8	69.9	3.2	7.0

Table 5: Overview of the noisy training splits in NOISEBENCH for German. The table shows the noise level, the micro-averaged token-level F1 score ($F1_{\text{token}}$), micro-averaged entity-level F1 ($F1_{\text{entity}}$), the number of entities (*Total*), number of correct entities (*Correct*) and share of each error type: missing mentions (*Missing (FN)*), non-entity mentions (*Non-entity (FP)*), wrong type (*Type*) and partial matches (*Partial*). All metrics are in comparison to the Clean split.

B Implementation Details

In all our experiments with noise-robust methods we use the same `xlm-roberta-large` transformer as in the FLERT baseline with a batch size of 32, except for L2R, for which we used a batch size of 16 due to VRAM constraints.

B.1 Confident Learning

We use regular transformer fine-tuning and obtain predicted probabilities for each sample in the training dataset using cross-validation. The number of folds is the only parameter in this approach and we performed a small search before choosing 3 folds. We use the implementation by Klie et al. (2023) to adapt this approach for NER by aggregating token-level predictions. We perform the final sample selection on the sentence level, training the model only using sentences that do not contain entities flagged as errors or have missing entities.

B.2 Co-Regularization

We perform a hyperparameter sweep as suggested by the authors, and choose the best performing ones on the validation set of the respective noise type.

B.3 BOND

In our experiments we found that limiting BOND's first stage of training to 1 epoch is not enough for optimal performance, hence why we rely on the findings reported by Tänzer et al. (2022) and stop the first stage after the first 3 epochs. The second stage is limited to 7 epochs in order to reproduce the same training length as in the FLERT baseline. We update the teacher model in the second stage of training every 2 epochs as suggested by the authors for the CoNLL-03 dataset, and use hard pseudo-labels in the second stage, which we found to outperform soft pseudo-labels in our experiments.

B.4 CrossWeigh

We ran the CrossWeigh framework with 5 folds and 3 iterations, because according to the ablation experiments ran by the authors, higher numbers did not bring significant performance improvements. For a fair comparison, we adjust the CrossWeigh framework to use transformer fine-tuning as a base model. For the final training run using the sample weights, we used the same FLERT approach as in the baseline.

B.5 L2R

We rely on the implementation provided by Zhu et al. (2023b), test out two meta-learning rates while keeping the model learning rate fixed at 5e-6, and perform the validation step every 0.1 epoch with a patience of 10 validation steps.

B.6 MSR

We used the implementation provided by the authors (Zhu et al., 2023a) and the hyperparameters they selected for CoNLL, as stated in their paper. For the German dataset, we used `xlm-roberta-base` as a multilingual model.

B.7 Upper Bound: Additional Clean Data

This upper bound assumes an additional small dataset with high-quality labels is available. We fix this number to 100 sentences, which are randomly chosen from the validation split (otherwise not used to train the models). This training setting first fine-tunes the baseline model for 10 epochs, and then continues fine-tuning only on the small clean dataset for 5 more epochs.

B.8 GPT3.5

To obtain an LLM-annotated variant of the training splits, for NOISEBENCH we used `gpt-3.5-turbo-0613`, while for German we used `gpt-3.5-turbo-0125`.

C Size of Oracle Subsets

In Table 6, the number of clean sentences in the oracle subset for each noisy variant is shown. These oracle subsets are used as performance upper bounds, as explained in Section 4.1.1.

	% of all sentences	Oracle subset size
NoiseBench split		
Clean	100.0	4879
Expert	92.6	4483
Crowd++	79.4	3786
Crowd	55.3	2554
Distant	59.7	2728
Weak	49.6	2294
LLM	38.2	1705
German split		
Clean	100.0	10824
Expert	81.6	8827
LLM	37.8	4095

Table 6: Details about the oracle subset used as an upper performance bound. The table shows the percentage of clean sentences and the absolute number, for each noise type.

D Baseline for Uniform Noise

Table 7 shows the results from Experiment 1 for uniform noise. We can see the the model is quite robust to uniform noise and that it results in higher test performance, when compared to real noise of the same level. The average difference in F1 scores in 17 percentage points, which is why we focus on the more realistic oracle class-dependent noise simulation method in the main results of Experiments 1 and 2.

	Real noise		Uniform noise		Δ
	%Noise	*F1*	*%Noise*	*F1*	*F1*
Clean	0	94.0 ±0.0	-	-	-
Expert	5.5	89.8±0.2	5.4	93.8±0.3	4,0
Crowd++	15.3	86.7±0.3	16.1	92.8 ±0.5	6.1
Crowd	36.6	70.5±0.6	36.7	88.4 ±0.3	17.9
Distant	31.3	70.8±0.1	31.7	90.0 ±0.5	19.3
Weak	40.4	65.9±0.4	42.2	91.7 ±0.2	25.8
LLM	45.6	62.6±0.4	47.3	89.7 ±0.6	27.1
Average		74.4±0.3		91.40.4	17.0

Table 7: F1 scores on the Clean Test split of the baseline FLERT approach, fine-tuned on different noisy variants of the training set. The scores are averages of 3 runs. The column Δ (difference) refers to the difference in F1 score on the test split when training on a dataset with real noise compared to uniform noise.

E Memorization of Crowd, Weak and LLM Noise

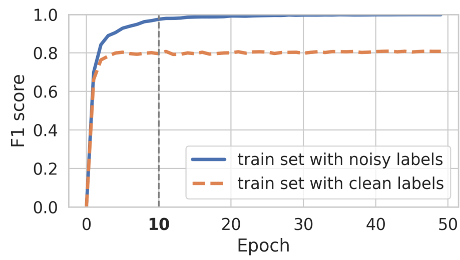
Figure 5 is an extension of Section 3.4 and shows the memorization plots from Experiment 2 for the Crowd, Weak and LLM dataset variants. We again observe immediate memorization of real noise and delayed memorization of simulated noise.

F Additional Experiments on Memorization

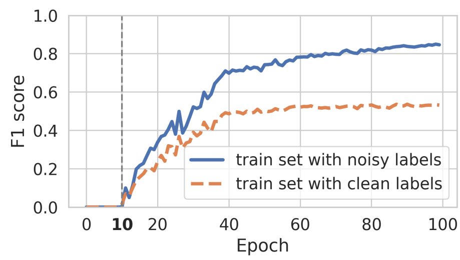
In addition to the main Experiment 2, we ran two ablation experiments regarding memorization.

F.1 Effect of Pre-training on Memorization

The first ablation compares fine-tuning a pretrained model and a model with randomly initialized weights. Figure 6 shows this comparison during an extended training run for the Crowd++ training variant, where we used DistilBERT (learning rate of 5e-05). We can see that even without pretraining, the model starts overfitting to the noisy labels and we can observe a large gap between the performance on the clean and noisy labels.



(a) Pre-trained model

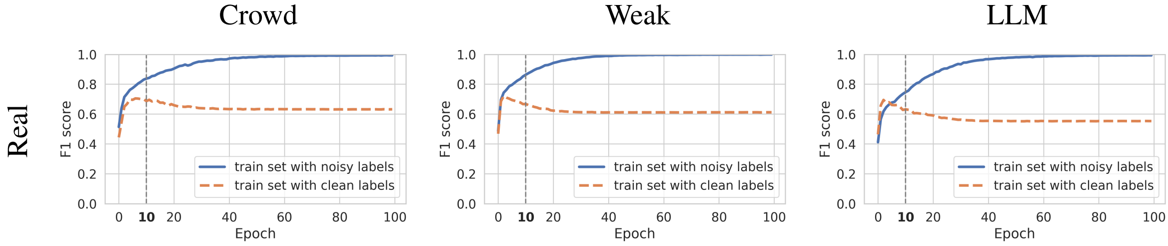


(b) Random initialization

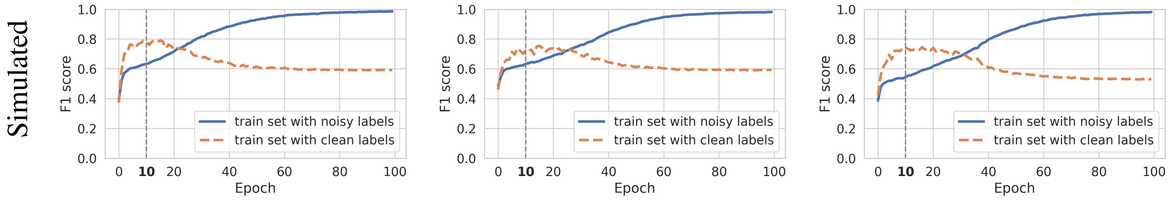
Figure 6: Memorization of label noise in DistilBert, using the pretrained model and a model with randomly initialized weights. The experiment was run for one noise type - Crowd++.

F.2 Memorization in a Smaller Model

The second experiment investigates memorization when fine-tuning a smaller model, DistilRoBERTa,



(a) Real Crowd - 36.6% noise (b) Real Weak - 40.4% noise (c) Real LLM - 44.6% noise



(d) Simul. Crowd - 41.3% noise (e) Simul. Weak - 41.2% noise (f) Simul. LLM - 47.2% noise

Figure 5: Comparison of model performance during extended training, for Crowd, Weak and LLM from NOISEBENCH. The top row shows models fine-tuned on label sets with real noise, while the bottom row models fine-tuned on a corresponding simulated noisy label set. The graphs show both the F1 score on the noisy training labels and on the clean training labels, for 3 different noise types. The plots are averages of 3 runs.

because the model we use in the main experiments, XLM-RoBERTa-Large, is quite large. Figure 7 shows an extended training run for three noise types in NoiseBench. We observe the same patterns of immediate and delayed memorization as with the larger model.

G Extended Performance Metrics

In this section we provide extended metrics of the predictive performance of the baseline FLERT method. These metrics and analysis correspond to Experiment 1 from Section 3.3.

G.1 Analysis of Test Errors

We can characterize the model predictions in a similar way as we characterized the different types of errors in Table 1 and Figure 1. Table 8 shows how representative different types of prediction errors are, expressed as a percentage of all errors. We can see that with Expert noise, a majority of the mistakes are wrong entity types. Furthermore, for the Crowd and Distant dataset versions, the largest number of errors is due to missing entities, while for the Weak and LLM datasets, the errors are mostly non-entities or wrong type. This is in line with the characteristics of the noisy datasets themselves, described in Table 1.

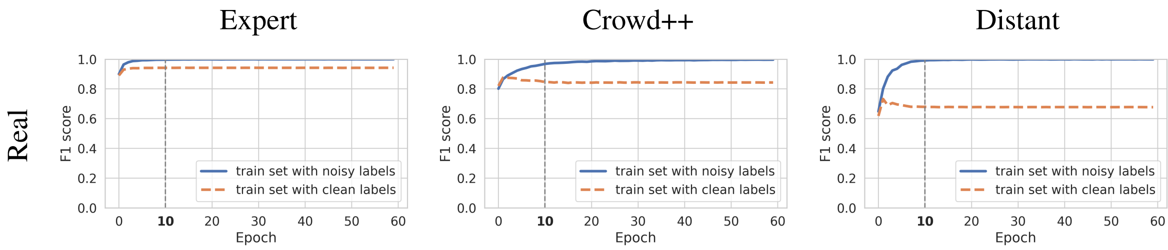
For German we make similar observations. For the clean variant, most errors are missing entities. However, for the two noisy variants, which include a large number of noisy non-entity annotations as seen in Table 5, the majority of prediction errors are also non-entity mentions.

We also examined the confusion matrices of the predictions. We were able to identify some patterns, regarding which types of errors are more prone to memorization. For most noise types in English NOISEBENCH, the largest number of prediction mistakes (out of the strings previously seen with a noisy label in the training set) were missing ORG and MISC entities, as well as ORG misclassified as LOC. These mistakes were present in a large number consistently across noise types.

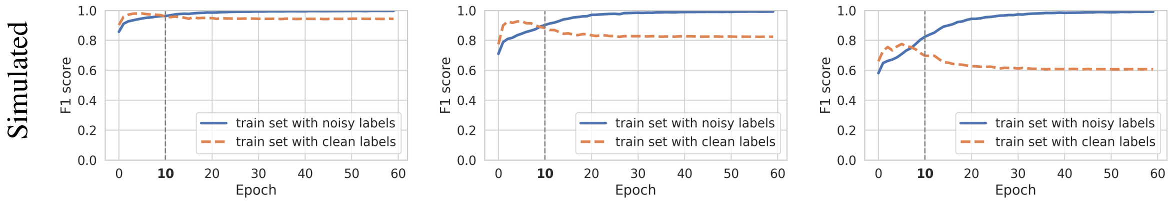
However, we also observed a large number of missing ORG and MISC entities in the predictions when using the clean training set, which indicates that this is an inherently difficult pattern, even when noise is not present. On another hand, the pattern of misclassifying ORG as LOC does not happen when clean data is available. Therefore we can conclude that when this type of noisy pattern is present in the training set, the models are not able to recognize it as noise and are not robust to it.

G.2 Per-Class Metrics

We provide per-class metrics for a more extensive evaluation of the performance of the baseline method, for both German and English, in Table 9.



(a) Real Expert - 5.5% noise (b) Real Crowd++ - 18% noise (c) Real Distant - 31.3% noise



(d) Simul. Expert - 6% noise (e) Simul. Crowd++ - 15% noise (f) Simul. Distant - 39% noise

Figure 7: Comparison of model performance during extended training with a smaller model, Distil-RoBERTa. The top row shows models fine-tuned on label sets with real noise, while the bottom row models fine-tuned on a corresponding simulated noisy label set. The graphs show the F1 score on the noisy training labels and on the clean training labels, for 3 noise types. The plots are averages of 3 runs.

	% Prediction errors			
	Missing	*Non-entity*	*Type*	*Partial*
NoiseBench split				
Clean	13.9	25.4	29.6	31.1
Expert	12.2	14.4	54.1	19.3
Crowd++	27.0	11.9	42.1	18.9
Crowd	55.2	6.4	25.3	13.1
Distant	64.6	8.2	14.4	12.8
Weak	14.2	24.5	49.5	11.8
LLM	20.5	34.9	38.3	6.4
German split				
Clean	39.9	26.4	16.7	16.9
Expert	17.8	61.9	11.7	8.6
LLM	26.7	61.5	4.8	6.9

Table 8: Overview of the percentage of different types of prediction errors

We can see that both precision and recall for the MISC class are generally lower than the other classes. This is especially noteworthy in the Expert label set of the German split, which does not have a high noise share, but it does have very low performance on MISC. This is however expected, as most of the noisy labels in this label set are related to MISC entities.

	LOC			ORG			PER			MISC		
	Prec.	*Recall*	*#Ent.*	*Prec.*	*Recall*	*#Ent.*	*Prec.*	*Recall*	*#Ent.*	*Prec.*	*Recall*	*#Ent.*
NoiseBench split												
Clean	93.9 ±0.6	93.5 ±0.4	1413	93.5 ±0.1	94.6 ±0.1	1909	99.0 ±0.2	99.1 ±0.1	1591	82.9 ±1.0	86.2 ±1.0	812
Expert	81.1 ±0.1	95.5 ±0.6	1413	92.0 ±0.6	81.9 ±0.3	1909	98.4 ±0.7	99.0 ±0.3	1591	85.3 ±0.9	81.4 ±1.1	812
Crowd++	75.9 ±0.6	94.4 ±0.4	1413	91.8 ±0.3	75.4 ±0.8	1909	97.6 ±0.5	97.8 ±0.3	1591	86.5 ±0.5	70.8 ±0.1	812
Crowd	65.5 ±0.4	90.7 ±1.2	1413	83.7 ±0.8	44.5 ±0.3	1909	93.6 ±1.9	71.7 ±1.2	1591	84.4 ±0.4	47.5 ±0.6	812
Distant	83.8 ±0.7	74.8 ±0.2	1413	85.3 ±1.3	55.9 ±1.0	1909	75.7 ±1.7	84.8 ±0.8	1591	98.6 ±1.4	13.8 ±3.1	812
Weak	52.5 ±0.2	93.1 ±0.1	1413	49.9 ±0.2	34.6 ±0.3	1909	88.5 ±0.3	88.9 ±1.0	1591	84.2 ±0.9	57.3 ±0.9	812
LLM	52.7 ±0.3	84.6 ±0.5	1413	57.7 ±0.9	45.4 ±0.5	1909	95.4 ±0.6	98.3 ±0.1	1591	12.7 ±0.4	11.8 ±0.4	812
German split												
Clean	93.0 ±0.7	90.8 ±0.2	1051	80.3 ±0.6	82.3 ±0.3	584	96.6 ±0.4	97.1 ±0.2	1210	77.1 ±1.7	56.5 ±0.6	206
Expert	88.4 ±0.6	83.3 ±0.9	1051	67.1 ±0.8	84.0 ±0.2	584	96.9 ±1.0	96.0 ±0.4	1210	13.1 ±0.3	40.6 ±0.8	206
LLM	63.9 ±0.4	70.3 ±0.9	1051	37.3 ±0.7	82.8 ±1.4	584	66.8 ±0.2	63.7 ±0.6	1210	10.4 ±0.6	18.4 ±1.6	206

Table 9: Per-class metrics of the predictions on the Clean test set