Which Examples Should be Multiply Annotated? Active Learning When Annotators May Disagree

Connor Baumler*
University of Maryland
baumler@cs.umd.edu

Anna Sotnikova*
University of Maryland
asotniko@umd.edu

Hal Daumé III
University of Maryland
Microsoft Research
me@hal3.name

Abstract

Linguistic annotations, especially for controversial topics like hate speech detection, are frequently contested due to annotator backgrounds and positionalities. In such situations, preserving this disagreement through the machine learning pipeline can be important for downstream use cases. However, capturing disagreement can increase annotation time and expense. Fortunately, for many tasks, not all examples are equally controversial; we develop an active learning approach, Disagreement Aware Active Learning (DAAL) that concentrates annotations on examples where model entropy and annotator entropy are the most different. Because we cannot know the true entropy of annotations on unlabeled examples, we estimate a model that predicts annotator entropy trained using very few multiply-labeled examples. We find that traditional uncertainty-based active learning underperforms simple passive learning on tasks with high levels of disagreement, but that our active learning approach is able to successfully improve on passive learning, reducing the number of annotations required by at least 24% on average across several datasets.

1 Introduction

Disagreement in annotations is natural for humans, often depending on one's background, identity, and positionality. This is especially salient when building classifiers for hate speech, toxicity, stereotypes, and offensiveness, where recent work has shown the importance of modeling annotator diversity and accounting for the full distribution of annotations rather than just a "majority vote" label (Plank, 2022; Sap et al., 2022; Uma et al., 2021a; Zhang et al., 2021b). However, collecting annotations in high-disagreement scenarios is expensive in time, effort, and money, because modeling annotator uncertainty may require collecting many labels for each example.

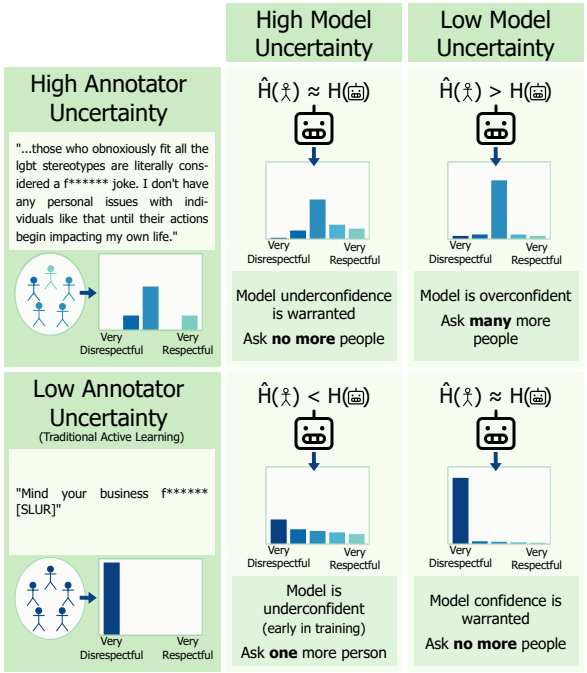


Figure 1: Utility of annotations when annotators disagree/agree (rows) and when the model is unconfident/confident (columns). When model uncertainty is well-calibrated with annotator uncertainty, no more annotations are needed. However, additional annotation(s) can be advantageous when the model is underconfident (e.g., uncertain on high agreement examples early in training) or overconfident (i.e., overly certain on high disagreement examples). Examples are edited to remove swears and slurs, and the high annotator uncertainty example is lightly paraphrased for anonymity.

To decrease labeling costs, we turn to active learning, a machine learning framework that *selectively* elicits annotations on examples that are most likely to improve a model's performance while minimizing annotation costs (Hanneke, 2014; Settles, 2009, i.a.). Many active learning approaches select examples to label based on some measure of *model uncertainty*, with the aim of driving down model uncertainty as quickly as possible.

*Equal contribution.

However, in the case of potential annotator disagreement, uncertainty-based sampling is not obviously a good strategy. Intuitively, an algorithm should collect annotations on examples for which the model uncertainty is significantly *different from* the annotator uncertainty, so these new annotations are able to help calibrate the model. Similarly, an active learning algorithm might plausibly request new labels on already labeled samples to better model the full distribution of possible annotations. This raises a "Goldilocks problem": on examples with complete annotator agreement, we do not need more than one annotation, while on examples with complete disagreement, no annotations are needed; it is precisely those examples in the middle—some, but not perfect agreement—on which multiple annotations are potentially useful.

In this paper, we develop DAAL (Disagreement Aware Active Learning),[1] an active learning algorithm for training classifiers to predict full label distributions on tasks with likely disagreement. DAAL first builds an *entropy predictor* that estimates, for a given example, how much annotator disagreement there is likely to be on that example. Then, using this entropy predictor, DAAL trains a *task predictor* that queries examples for which the current task predictor's current entropy is most *different from* its estimated human entropy (Figure 1). We evaluate DAAL on several text classification problems related to English hate speech and toxicity detection, finding that:

1. Traditional uncertainty-based active learning algorithms *under-perform* pure random sampling, especially on tasks with high annotator disagreement, and especially when the goal is to estimate the full label distribution (rather than just the majority vote label);
2. It is possible to estimate a high quality entropy predictor using a much smaller number of samples than is needed to learn the task predictor, making DAAL a feasible approach.
3. DAAL can effectively reduce the number of needed annotations by at least 24% on average to achieve the same predictive performance, in comparison to the strongest competitor.
4. DAAL automatically *selectively* re-annotates the same example multiple times, and also sometimes re-annotates examples specifically to *increase* the task predictor's uncertainty, both typically during later phases of learning.

[1] https://github.com/ctbaumler/daal

2 Related Work

Data collection has always been a challenge in NLP, especially for subjective and ambiguous topics such as stereotypes, biases, hate speech, and toxicity. It has been shown that examples annotators disagree on can be valuable inputs to classifiers, and that disagreement is more than just noise (Basile et al., 2021; Leonardelli et al., 2021; Larimore et al., 2021; Pavlick and Kwiatkowski, 2019; Palomaki et al., 2018). Moreover, having a diverse annotator pool can be crucial to performance (Almanea and Poesio, 2022; Akhtar et al., 2021; Sotnikova et al., 2021). Baan et al. (2022) and Plank (2022) demonstrate that, when the goal is to produce full label distributions, evaluating classifiers against the majority vote can give misleading results. Both argue that dataset developers should release un-aggregated labels with datasets. Recent approaches to learning to predict full-label distributions—rather than just majority vote labels—often train on "soft labels," treating each annotation as a separate example, instead of majority vote labels (Davani et al., 2022; Fornaciari et al., 2021; Uma et al., 2021b; Klenner et al., 2020; Aroyo and Welty, 2013).

One of the most commonly deployed approaches to minimize the number of collected annotations to train a model is active learning, where the main idea is to collect only those annotations that might be helpful for improving model performance. Active learning algorithms operate iteratively, where in each round a small number (often one) of examples are requested to be annotated. These annotated examples are added to a training set, a model is trained on that dataset, and then the process repeats. One popular strategy for selecting which examples to have annotated in each round is uncertainty sampling, where the model queries on examples on which it is the least certain (Ramirez-Loaiza et al., 2017; Culotta and McCallum, 2005; Lewis, 1995), with uncertainty often measured by the current entropy of the label distribution produced by the model at the current round.

3 Learning with Annotator Disagreement

In this section, we motivate and formalize the problem we aim to solve, describe passive and active learning baselines, and introduce our algorithm, DAAL (Disagreement Aware Active Learning).

3.1 Motivation

When considering a task and dataset with (potential) annotator disagreement, we aim to capture this disagreement by training a classifier that predicts a full-label distribution, rather than a single label. When classifiers are part of a larger system, predicting full-label distributions enables classifier uncertainty to be used directly in that system, for instance to trade-off false positives and false negatives under deployment-specific cost models.

Beyond simply learning a classifier that can predict label distributions, we also aim to minimize the number of samples annotated. There are standard reasons for doing so, namely that annotation costs time and money. Beyond that, however, annotation of data related to hate speech, toxic language, and related tasks, comes with an additional burden to annotator mental health. And so we also wish to minimize the burden on annotators.

3.2 Task Definition

To formalize the task at hand, let X be an input space (e.g., over social media posts), Y be an output space (e.g., over levels of toxicity), and let $\Delta(Y)$ be the space of distributions over Y (i.e., distribution over toxicity levels, possibly obtained by querying multiple annotators).

The learning problem is defined by a fixed but unknown distribution $P_X(x)$ over X—representing the sampling distribution of inputs—and an oracle labeling distribution $P_{Y|X}(y|x)$ over labels y given an input x, where the distribution reflects the fact that different annotators may provide different labels. In general, the learning goal is to learn a task predictor $f_\theta : X \to \Delta(Y)$ that minimizes an expected loss over xs drawn from P_X and labels drawn from $P_{Y|X}$ given that x. Because we are interested in predicting a soft label distribution, and not a single label, we measure loss using a distribution measure: Jensen-Shannon divergence between $P_{Y|X}$ and f_θ on each x:

$$\mathcal{L}(f_\theta) = \mathbb{E}_{x\sim P_X} \text{JS}\left(P_{Y|X}(\cdot|x), f_\theta(x)\right) \quad (1)$$

$$\text{JS}(p_1, p_2) = \tfrac{1}{2}\big(\text{KL}(p_1||\bar{p}) + \text{KL}(p_2||\bar{p})\big) \quad (2)$$

$$\text{where } \bar{p}(z) = \tfrac{1}{2}\big(p_1(z) + p_2(z)\big)$$

The active learning variant of this problem supposes that we have access to a pool of unlabeled data $U \subset X$ sampled from P_X, a query budget B, as well as query access to $P_{Y|X}$: given an x, we can draw a single label $y \sim P_{Y|X}(\cdot|x)$, at a cost. The task is: given U, B, and sample access to $P_{Y|X}$, learn a soft classifier $f_\theta : X \to \Delta(Y)$ that minimizes Eq 1 using at most B queries to $P_{Y|X}$.

3.3 Passive Learning Baseline

The simplest approach to learning a classifier in the framework described in the previous subsection is passive learning: pick a random subset of examples from U, label them all, and train a classifier on the resulting dataset. There is, however, a subtlety in the disagreement case even for passive learning: is it better to select B examples and to query $P_{Y|X}$ once for each one, or is it better to select B/N examples and to query $P_{Y|X}$ N times for each?[2] We consider both modes, which we refer to as "single" (one at a time) and "batched" (N at a time).

Formally, passive learning first selects a pool $D_X \subset U$ uniformly at random of size B/N, and, for each $x \in D$, queries $P_{Y|X}(\cdot|x)$ independently N times to obtain labels $y_1^{(x)}, \ldots, y_N^{(x)}$. Following standard practice (see § 2), we then construct a labeled dataset $D = \{(x, y_n^{(x)}) : x \in D_X, 1 \leq n \leq N\}$ and train a classifier f_θ on D.

3.4 Entropy-Based Active Learning Baseline

Entropy-based active learning repeatedly queries the oracle $P_{Y|X}$ each round, selecting an example for annotation based on the entropy of the current classifier. This is formally specified in Alg. 1. At each of B rounds, a single example x_b is selected as the one on which the current classifier has maximum uncertainty. This example is then given to the oracle $P_{Y|X}$ and a label y_b is sampled. This labeled example is added to the dataset D and the process repeats. Similar to passive learning, entropy-based active learning can be run either in "single" mode (one annotation at a time) or "batched" (N at a time).

In practice, entropy-based active learning can be computationally infeasible: training a new classifier after every new sample is costly, and re-evaluating the entropy of all of U after every new sample is also costly. To reduce this computational cost—at the price of some loss in performance—we only retrain the classifier and re-evaluate entropy every 10 rounds. (This is equivalent to selecting the 10 examples with the highest entropy in each round.)

[2] This conundrum applies even in the setting without disagreement because of label noise and has been studied theoretically (Khetan et al., 2018) and empirically (Zhang et al., 2021a; Dong et al., 2021).

Algorithm 1: Entropy-Based AL

Input: Unlabeled data U, budget size B

$D_1 \leftarrow \{\}$
for $b = 1 \ldots B$ **do**
  $f_\theta \leftarrow$ task classifier trained on D_b
  $x_b \leftarrow \arg\max_{x \in U} H(\, f_\theta(x)\,)$
  $y_b \sim P_{Y|X}(\cdot|x_b)$ – query oracle
  $D_{b+1} \leftarrow D_b \cup \{(x_b, y_b)\}$
return f_θ

3.5 Our Approach: Disagreement Aware Active Learning

The intuition behind entropy-based active learning is that driving down the entropy of f_θ is a good idea and that the most effective way to drive down that entropy is to elicit labels on samples on which f_θ currently has high entropy. Unfortunately, while entropy-based active learning has been incredibly effective at reducing labeling cost on relatively unambiguous labels, we find that it often performs *worse* than passive learning on tasks where annotators disagree (§ 5.1). This likely happens because when the goal is to predict a label distribution, and the ground truth entropy of that distribution is non-zero, then attempting to drive the entropy of f_θ to zero is potentially misguided.

Consequently, we need a new approach that treats annotator uncertainty as a first-class citizen. To gain an intuition of what such an algorithm should do, consider an example where annotators agree. Here, new labels will be the same as existing labels and thus only reinforce the model's predictions when added to training data. For an example where annotators disagree, new labels will potentially be quite different. When a newly sampled label is surprising given the model's current predicted label distribution, this will increase the model's belief in the new label and decrease the model's certainty.

Querying based on different levels of annotator uncertainty can affect model confidence, but this is only necessary when the model's level of confidence is incorrect. If the model is certain on an example that annotators agree on, then this is a warranted level of confidence, and there is no need to reinforce the correct distribution with more labels. In the opposite case, the model's uncertainty on an example where humans disagree is justified, so even if collecting more annotations could help increase model certainty, this would be undesirable.

Therefore, the useful examples to query on are those with a *mismatch* between the level of annotator uncertainty and model uncertainty, rather than just high model uncertainty. This suggests a variation of entropy-based active learning (Alg. 1) in which x_b is selected not to maximize model uncertainty, $H(f_\theta(x))$ but to maximize the *difference* between model uncertainty and human uncertainty:

$$\arg\max_{x \in U} |H(\underbrace{f_\theta(x)}_{\text{Task model's predicted label dist. on } x}) - H(\underbrace{P_{Y|X}(\cdot|x)}_{\text{Ground truth label distribution on } x})| \quad (3)$$

Unfortunately, we cannot compute Eq 3 because we do not know $H(P_{Y|X}(\cdot|x))$ and to estimate it would require querying $P_{Y|X}$ multiple times—exactly what we are trying to avoid. To address this, DAAL trains an *entropy predictor* that estimates $H(P_{Y|X}(\cdot|x))$ for any x, and uses this estimated entropy in place of the true entropy in Eq 3. Fortunately, we find that this entropy predictor can be trained with a sufficiently small number of samples so as not to overshadow the benefits of using active learning (see §5.3).

Our proposed algorithm is detailed in Alg. 2. In the beginning, DAAL builds an initial dataset for estimating an entropy predictor by querying N annotations for B_{ent} random samples, similar to passive learning. This entropy predictor is a regressor trained to predict the observed empirical entropy of those N annotations given an input x. The remainder of DAAL is parallel to entropy-based active learning (Alg. 1). In each round, an example is selected based on the absolute difference between model entropy and *estimated* human entropy:

$$x_b = \arg\max_{x} |H(\underbrace{f_\theta(x)}_{\text{Task model's predicted label dist. on } x}) - \underbrace{f_{\text{ent}}(x)}_{\text{Predicted annotator entropy on } x}| \quad (4)$$

Every time DAAL queries for more annotations, a new f_θ is trained from scratch, and the procedure is repeated until the annotation budget is exhausted. If needed, DAAL may query the same

Algorithm 2: DAAL

Input: Unlabeled data U, budget size B, entropy-predictor budget B_{ent} and number of entropy annotations N

$D_X \leftarrow B_{\text{ent}}$ random samples from U
for $x \in D_X$, $n = 1 \ldots N$, sample $y_n^{(x)} \sim P_{Y|X}(\cdot|x)$
$D_H \leftarrow \{(x, H(\{y_n^{(x)}\}_{n=1}^N) : x \in X\}$
$f_{\text{ent}} \leftarrow$ entropy predictor trained on D_H
$D_1 \leftarrow \{(x, y_n^{(x)}) : x \in X, n = 1 \ldots N\}$
for $b = 1 \ldots B - B_{ent} \times N$ **do**
  $f_\theta \leftarrow$ task classifier trained on D_b
  $x_b \leftarrow \arg\max_{x \in U} |H(f_\theta(x)) - f_{\text{ent}}(x)|$
  $y_b \sim P_{Y|X}(\cdot|x_b)$
  $D_{b+1} \leftarrow D_b \cup \{(x_b, y_b)\}$
return f_θ

Characteristics	Measuring Hate Speech			Wikipedia	
	`Respect`	`Dehumanize`	`Genocide`	`Toxicity`	`Toxicity-5`
Number of Total Examples	$17,282$	$17,282$	$17,282$	$20,000$	$20,000$
Avg Number of Annotations per Example	3.35	3.35	3.35	10.0	5.0
Number of Examples Test Set	$1,778$	$1,778$	$1,778$	$2,000$	$2,000$
Probability Two Annotators Disagree	0.520	0.689	0.371	0.524	0.522

Table 1: Dataset statistics for MHS and Wikipedia tasks.

examples multiple times, but it is not required to waste the annotation budget on examples where all useful information is learned after one annotation (or zero). When the annotator entropy is zero (i.e., all annotators agree on a single label), DAAL reduces to simple uncertainty sampling. As in the case of entropy-based active learning, retraining f_θ and recomputing model entropy after every sample is computationally expensive, so in practice, we retrain and re-evaluate only after every 10 rounds.

4 Experimental Setup

In this section, we introduce the datasets we use and experimental details.

4.1 Datasets

We conduct experiments in simulation by starting with datasets with multiple annotations per example and returning one of these at random when the oracle is called. We choose two datasets with multiple labels for each attribute: Measuring Hate Speech (MHS) (Sachdeva et al., 2022) and Wikipedia Talk (Wulczyn et al., 2017); basic data statistics are summarized in Table 1.

The MHS dataset was collected from YouTube, Twitter, and Reddit examples. It has nine scale attributes that contribute to their definition of hate speech, from which we select three for our experiments: `Dehumanize` (which has high levels of human disagreement), `Respect` (which has medium levels), and `Genocide` (which has low levels). Each attribute is labeled for every example on a five-point Likert scale from strongly disagree to strongly agree. There are $50k$ examples, each of which is annotated between 1 and 6 times in the main dataset (see Figure 17); for our simulated experiments we only consider those with $3-6$ annotations, resulting in around $20k$ total examples.

The Wikipedia dataset was created as a result of the Wikipedia Detox Project.[3] It has three attributes of which we select one for experiments—`Toxicity`—which is also rated on a five-point Likert scale from very toxic to very healthy. This data consists of $100k$ examples with 10 annotations per example in almost all cases; we randomly downselect to $20k$ examples for congruity with MHS.

4.2 Experimental Details

We measure the classifier's performance according to Jensen-Shannon divergence (JS), defined in Eq 2.[4] We introduce an oracle trained on the full dataset for each task to calibrate model performance against the best possible.

For each method, we finetune RoBERTa-base (Liu et al., 2020). We finetune the task model each round from scratch, which worked better than continuing training in preliminary experiments. We use early stopping with a tolerance of 1 based on the KL divergence between the model's predicted distribution and the distribution of annotator votes on a held-out set, training for a maximum of 50 epochs. For DAAL's entropy predictor, we also finetune a RoBERTa-base model and use early stopping with a tolerance of 5 based on the mean squared error on the held-out set.

Each experiment's result is averaged over 5 runs, and we present 95% confidence intervals based on these runs. For all algorithms, we disallow querying on examples where all available annotations are already in the training set.[5]

5 Results and Analysis

In this section, we present results for baseline methods (§5.1) and DAAL (§5.2). We also investigate how the budget size and the number of annotations per example affect the entropy predictor's performance (§5.3). In addition, we discuss in which sit-

[3] `https://meta.wikimedia.org/wiki/Research:Detox/Data_Release`

[4] We additionally report total variational distance as well as Macro F1 and accuracy in the Appendix.

[5] This issue only arises in simulation: in a real condition, one could always query more. In practice, we found that re-annotation queries were not frequent enough to raise concerns.

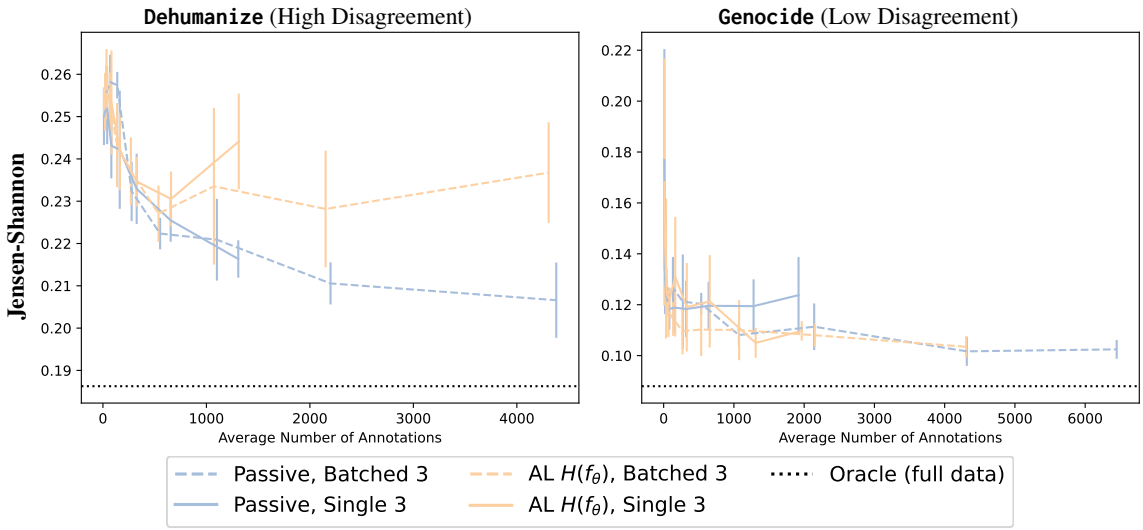


Figure 2: JS divergence scores for two attributes from the MHS dataset for passive learning baselines and entropy-based active learning (AL) baselines. For these experiments, we define $N \approx 3$, which means that there are approximately 3 annotations per example available in the data pool.[6] Both baselines have two variations when querying: "Batched" receives all 3 annotations per example while "Single" receives only one.

uations the models request additional annotations for already-seen examples over new ones (§5.4).

5.1 How Do Levels of Disagreement Impact Baselines?

To start, we seek to understand how levels of disagreement impact the efficacy of passive and active learner baselines. To do this, we compare high and low disagreement attributes (`Dehumanize` and `Genocide`). Learning curves on these tasks are shown in Figure 2. First, we see that the level of disagreement affects which approach is more effective. When annotators generally agree—as in `Genocide`—the active learner works well, outperforming passive learning for a distribution measure, JS divergence (Figure 2, right). Second, we see that on the high disagreement attribute (`Dehumanize`), active learning is worse than passive learning by a significant gap (Figure 2, left). We find a similar but weaker effect on accuracy-based measures in §A.1. We also show that using hard labels significantly hurts baseline performance on our task in §A.2.

In Figure 2, we can also compare the "batched" mode (when the model queries examples with $N = 3$ annotations simultaneously) and the "single" mode (when the model queries annotations individually). We can see that, for the low disagreement attribute, "single" active learning achieves comparable JS to "batched", but on average requires fewer annotations to reach the minimum. For the high disagreement attribute, the trend is less clear, but in the next section, we show that indeed querying a single annotation at a time is more effective for DAAL.

	Passive		Active $H(f_\theta)$	
Dataset	**Batch**	**Single**	**Batch**	**Single**
`Dehumanize`	2.05	1.80	> 7.60	> 2.32
`Respect`	1.44	1.25	3.52	> 1.47
`Genocide`	> 4.20	> 1.25	> 2.80	> 1.28
`Toxicity`	1.46	> 1.20	0.97	> 1.32
`Toxicity-5`	> 4.18	> 1.25	0.90	> 1.36
Average	> 2.67	> 1.35	> 3.16	> 1.55

Table 2: How many times more annotations the baselines require to achieve the same JS as DAAL.

5.2 Is DAAL Effective at Learning Distributions?

To compare results with the baselines, for each task we select the single strongest baseline from passive learning and entropy-based active learning to compare against.[7] We measure improvement in terms of the number of annotations needed for the model to achieve within 5% of its best possible JS divergence. Results are in Figure 3 and Table 2.

As we can see in Figure 3, DAAL achieves competitive JS on fewer annotations on aver-

[6] As discussed in §4.1, we use a portion of the MHS dataset that does not have a consistent number of annotations per example. For simplicity, we report results on this dataset as $N = 3$ as nearly $\frac{2}{3}$ of examples had 3 annotations.

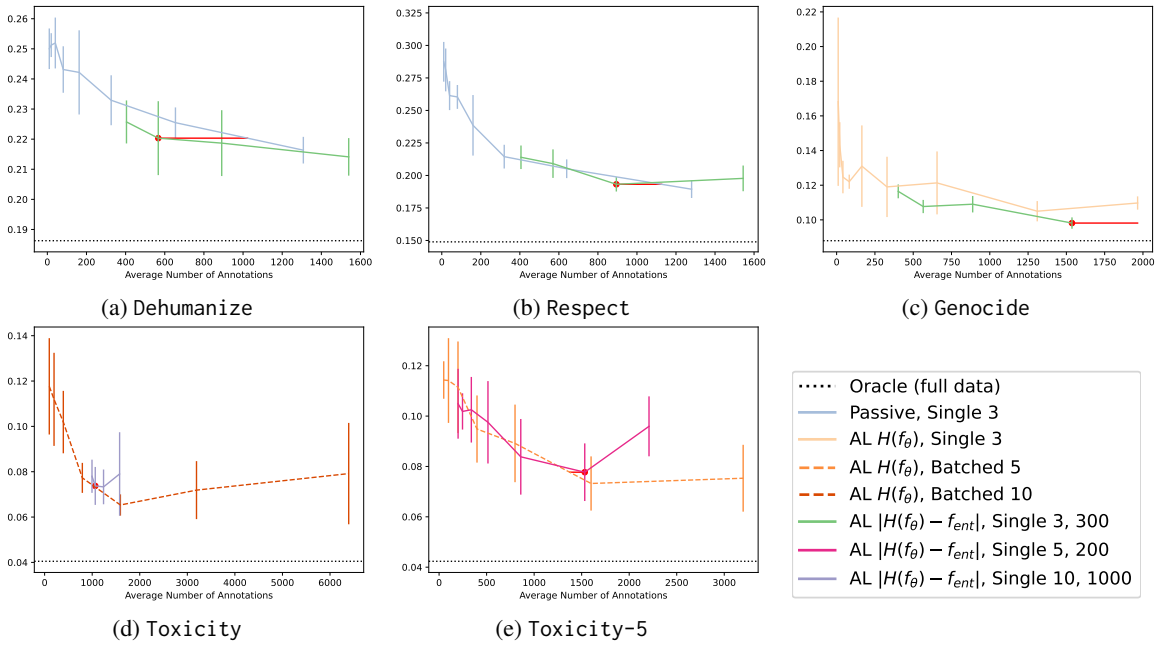


Figure 3: Jensen-Shannon divergence vs the number of required annotations. The lines in red show DAAL's improvement in the number of annotations. They connect the first measurement where DAAL was within 5% of its best JS to the point where the baseline achieves the same performance (if available). We compare DAAL with the empirically determined best budget size (See §5.3) and best performing baseline. We show in the legend labels whether the task model receives single or batched annotations for queried examples, the number of available annotations per example, and (for DAAL) the size of the entropy predictor's budget in annotations. The x-axis includes the annotations in the entropy predictor's budget.

age than all baselines. Other approaches might achieve the same performance but require at least 26% more annotations on average. For instance, DAAL achieves 0.225 JS divergence for the `Dehumanize` attribute after approximately 566 annotations, while the best baseline needs 1022 annotations to achieve the same performance (80% more). The one exception is on the `Toxicity` dataset, which we explore in §5.3.

In some cases, as with the `Genocide` attribute, the baseline models never get to the same performance as DAAL. We observe no strong pattern for DAAL working better or worse for high versus low disagreement attributes, suggesting that it's a "safe" option that can also be used in more traditional learning settings where there may not be much disagreement.

5.3 Size of the Entropy Budget, B_{ent}

We explore different budgets for the annotator entropy predictor described in §3.5. We experiment with budgets of 25, 100, and 200 examples on MHS `Respect`. Since the entropy predictor must be trained on multiply-annotated examples, our goal is to ensure it can be trained with a very small budget. The comparison of performances is shown in Figure 4. In general, we see that the entropy predictor can, indeed, be learned with relatively few examples and that a budget of 100 examples is near optimal. We confirm that this finding extends to the `Toxicity` dataset in §A.4.

In §5.2, we noted a situation on the `Toxicity` dataset when DAAL performs slightly worse (requires about 4% to 11% more annotations) than entropy-based active learning (Table 2). This dataset has markedly more annotations per example (Table 1), which is an artifact of the simulation used for the experiment. For a direct comparison, we repeat this experiment where we fix the total number of annotations to smaller values. Results

[7] Beyond the two simple active and passive learning baselines discussed in §3.3 and §3.4, we also considered BADGE (Ash et al., 2020), an active learning method that samples a diverse set of uncertain examples to annotate based on the magnitude of the gradient in the final hidden layer. Using BADGE's default hyperparameters and with 200 epochs per round (vs a limit of 50 for DAAL and the other baselines), we found that with both BERT and RoBERTa BADGE never outperformed our other baselines on datasets with annotator disagreement. For example, the final JS divergence of BADGE was 28% worse that the strongest baseline on MHS `Respect`, and 7% worse on MHS `Dehumanize`.

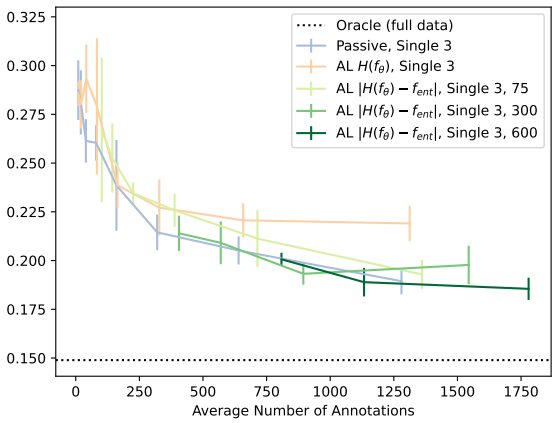


Figure 4: Comparison of JS Divergence when using different budgets for annotator entropy predictors described in § 3.5 on the MHS `Respect` attribute. We compare budgets of $25, 100$, and 200 examples with pre-collected annotations. For MHS ($N = 3$), this translates to budget sizes of $75, 300$, and 600 annotations

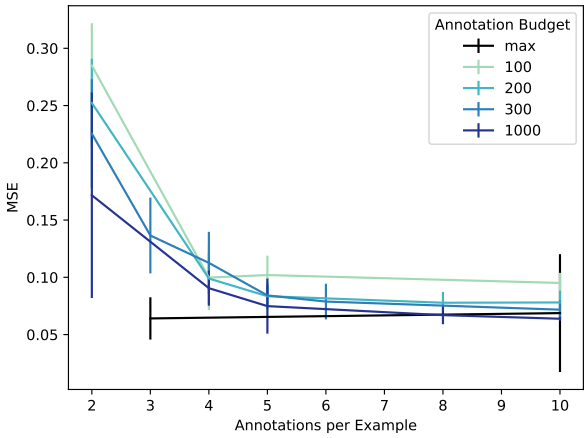


Figure 5: Entropy predictor performance on `Toxicity` on varying the total annotation budget and the number of annotations per example. We find that decreasing the annotations per example to 5 and the budget to 200 is generally sufficient.

are shown in Figure 5. We see that having more annotations per example gives better performance on the entropy predictor. (We show task model results on 3, 5, and 10 annotation per example DAAL in §A.4.) We notice that the optimal number of annotations is 5 per example, which suggests 5 might be a reasonable cap for the maximum number of times a single example could be queried in a real-world deployment.

5.4 f_{ent} vs $H(f_\theta)$ and Re-annotation Strategy

DAAL chooses examples to query based on the absolute difference between model and annotator entropy (See §3.5). This means that the model can select two kinds of examples depending on which term is larger. When $H(f_\theta) > f_{ent}$, the model is unsure of the correct label but predicts that annotators will agree on the label. When $f_{ent} > H(f_\theta)$, the model is overconfident in its label prediction given its prediction of annotator agreement levels.

In Figure 6, we consider which of these two kinds of examples the model is querying on at different points in learning. We find that our model begins by querying overwhelmingly on cases with $H(f_\theta) > f_{ent}$ but that the reverse is true later in training. This can be interpreted as beginning with "easy" examples where annotators are likely to agree and then choosing examples with higher disagreement later to correct overconfidence.

We also consider how often DAAL re-annotates an already annotated example. In Figure 7, we see that early in training, DAAL mostly chooses to query on new examples, but in the second half, about $2/3$ of annotations are re-annotations.

Combining this change in re-annotation rate with the change in which term dominates the query function, we can see a more clear strategy. Early in training, when the model is focusing on examples with low f_{ent}, there is no need to query for multiple labels. Once the model starts considering more examples with high f_{ent}, re-annotations become necessary to better capture the annotator distribution. These re-annotations are largely not given to examples with low f_{ent}, as these are not likely to require more than one annotation.

6 Conclusion

In this paper, we emphasize the importance of accounting for disagreement present in data. We propose DAAL, an active learning approach, which incorporates both annotator and model uncertainties, aiming to reduce the cost of annotation. This cost includes both time and money, but also an often overlooked cost related to the repeated exposure of annotators to toxic and harmful content. When the annotation is performed on crowdsourcing platforms, where workers are often from vulnerable populations who may require more flexible employment options—such as those with disabilities or who have caregiver roles (Berg, 2016)—this mental health cost compounds existing marginalization.

In our experiments on training classifiers for hate speech and toxicity detection, we show that DAAL achieves comparable Jensen-Shannon di-

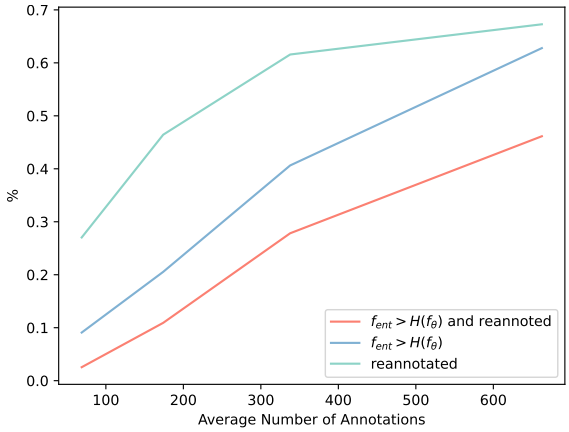


Figure 6: Re-annotation rate and f_{ent} vs $H(f_\theta)$ strategy for DAAL on `Toxicity`. Like Figure 7, the re-annotation rate increases over time (green). Additionally, the selection strategy goes from choosing mostly examples where $f_{ent}(x) \leq H(f_\theta(x))$ to choosing the opposite (blue). Later in training, these increased re-annotations largely go to examples where $f_{ent}(x) > H(f_\theta(x))$ (red).

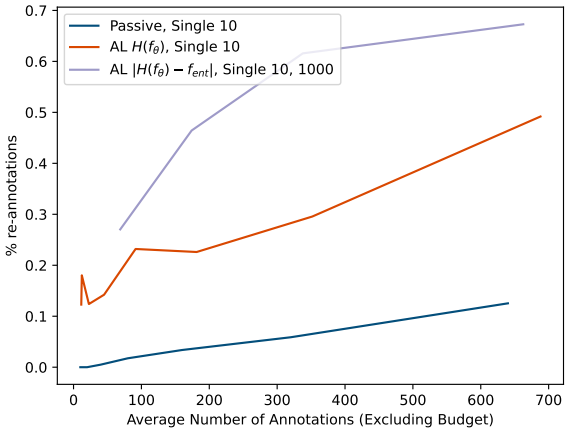


Figure 7: Re-annotation rate for single annotation strategies on `Toxicity`. We find that our method has a consistently higher re-annotation rate than the baselines and that the rate increases over time.

vergence with the classic baselines' performance but requires an average of $1.235\times$ fewer annotations in the worst case. It is also equally effective when there is little annotator disagreement, making it a strong general solution candidate even when one does not know ahead of time how much annotator disagreement is likely for a given task.

7 Limitations

There are several limitations to our experiments: we work only with English data and with datasets concerning hate speech and toxicity. Frequently such data do not represent i.i.d. samples from the data that we might encounter in real life. In addition, experiments are all conducted in the simulation with these existing datasets. The annotations in the simulated experiments were already checked for quality by the original dataset creators (Sachdeva et al., 2022; Wulczyn et al., 2017). In real-world deployment, further steps would need to be taken to ensure that the entropy in annotations truly comes from disagreements and not other kinds of noise.

While DAAL is designed to capture disagreement due to annotator positionalities, the datasets used may not have had a diverse enough pool of annotators to fully test this. In the portion of the MHS dataset used in our experiments, 67.9% of annotators were cisgender, straight, and white, while only 0.4% of examples targeted this same population. The Wikipedia Talk dataset does not provide demographic information about its annotators.

A classifier for toxic text or hate speech trained on a pool of annotators whose backgrounds do not reflect anywhere near the full diversity of human identities (and especially the identities of the targets of the text being classified) is inherently limited. Applying such a classifier, whether it predicts a single label or a distribution, to text from and about marginalized populations not represented in the annotator pool carries inherent risks to the well-being of these populations. Such a classifier could systematically fail to flag content that annotators from privileged groups do not find harmful or incorrectly flag innocuous speech written by members of marginalized groups.

8 Acknowledgements

The authors are grateful to all the reviewers who have provided helpful suggestions to improve this work, and thank members of the CLIP lab at the University of Maryland for the support on this project.

References

Sohail Akhtar, Valerio Basile, and Viviana Patti. 2021. Whose opinions matter? perspective-aware models to identify opinions of hate speech victims in abusive language detection.

Dina Almanea and Massimo Poesio. 2022. ArMIS - the Arabic misogyny and sexism corpus with annotator subjective disagreements. In *Proceedings of the Thirteenth Language Resources and Evaluation*

Conference, pages 2282–2291, Marseille, France. European Language Resources Association.

Lora Aroyo and Chris Welty. 2013. Crowd truth: Harnessing disagreement in crowdsourcing a relation extraction gold standard. *WebSci2013. ACM*.

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

Joris Baan, Wilker Aziz, Barbara Plank, and Raquel Fernandez. 2022. Stop measuring calibration when humans disagree. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1892–1915, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Valerio Basile, Michael Fell, Tommaso Fornaciari, Dirk Hovy, Silviu Paun, Barbara Plank, Massimo Poesio, and Alexandra Uma. 2021. We need to consider disagreement in evaluation. In *Proceedings of the 1st Workshop on Benchmarking: Past, Present and Future*, pages 15–21, Online. Association for Computational Linguistics.

Janine Berg. 2016. Income security in the on-demand economy: Findings and policy lessons from a survey of crowdworkers. *Comparative Labor Law & Policy Journal*, 37(3).

Aron Culotta and Andrew McCallum. 2005. Reducing labeling effort for structured prediction tasks. In *AAAI*, volume 5, pages 746–751.

Aida Mostafazadeh Davani, Mark Díaz, and Vinodkumar Prabhakaran. 2022. Dealing with disagreements: Looking beyond the majority vote in subjective annotations. *Transactions of the Association for Computational Linguistics*, 10:92–110.

Xinyue Dong, Shilin Gu, Wenzhang Zhuge, Tingjin Luo, and Chenping Hou. 2021. Active label distribution learning. *Neurocomputing*, 436:12–21.

Tommaso Fornaciari, Alexandra Uma, Silviu Paun, Barbara Plank, Dirk Hovy, and Massimo Poesio. 2021. Beyond black & white: Leveraging annotator disagreement via soft-label multi-task learning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2591–2597, Online. Association for Computational Linguistics.

Steve Hanneke. 2014. Theory of disagreement-based active learning. *Foundations and Trends® in Machine Learning*, 7(2-3):131–309.

Ashish Khetan, Zachary C Lipton, and Animashree Anandkumar. 2018. Learning from noisy singly-labeled data. In *International Conference on Learning Representations*.

Manfred Klenner, Anne Göhring, and Michael Amsler. 2020. Harmonization sometimes harms. In *Proceedings of the 5th Swiss Text Analytics Conference (SwissText) & 16th Conference on Natural Language Processing (KONVENS)*.

Savannah Larimore, Ian Kennedy, Breon Haskett, and Alina Arseniev-Koehler. 2021. Reconsidering annotator disagreement about racist language: Noise or signal? In *Proceedings of the Ninth International Workshop on Natural Language Processing for Social Media*, pages 81–90, Online. Association for Computational Linguistics.

Elisa Leonardelli, Stefano Menini, Alessio Palmero Aprosio, Marco Guerini, and Sara Tonelli. 2021. Agreeing to disagree: Annotating offensive language datasets with annotators' disagreement. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10528–10539, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

David D. Lewis. 1995. A sequential algorithm for training text classifiers: Corrigendum and additional data. *SIGIR Forum*, 29(2):13–19.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Ro{bert}a: A robustly optimized {bert} pretraining approach.

Jennimaria Palomaki, Olivia Rhinehart, and Michael Tseng. 2018. A case for a range of acceptable annotations. In *Workshop on Subjectivity, Ambiguity and Disagreement in Crowdsourcing*.

Ellie Pavlick and Tom Kwiatkowski. 2019. Inherent disagreements in human textual inferences. *Transactions of the Association for Computational Linguistics*, 7:677–694.

Barbara Plank. 2022. The "problem" of human label variation: On ground truth in data, modeling and evaluation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10671–10682, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Maria E. Ramirez-Loaiza, Manali Sharma, Geet Kumar, and Mustafa Bilgic. 2017. Active learning: An empirical study of common baselines. *Data mining and knowledge discovery*, 31(2).

Pratik Sachdeva, Renata Barreto, Geoff Bacon, Alexander Sahn, Claudia von Vacano, and Chris Kennedy. 2022. The measuring hate speech corpus: Leveraging rasch measurement theory for data perspectivism. In *Proceedings of the 1st Workshop on Perspectivist Approaches to NLP @LREC2022*, pages 83–94, Marseille, France. European Language Resources Association.

Maarten Sap, Swabha Swayamdipta, Laura Vianna, Xuhui Zhou, Yejin Choi, and Noah A. Smith. 2022. Annotators with attitudes: How annotator beliefs and identities bias toxic language detection. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5884–5906, Seattle, United States. Association for Computational Linguistics.

Burr Settles. 2009. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison.

Anna Sotnikova, Yang Trista Cao, Hal Daumé III, and Rachel Rudinger. 2021. Analyzing stereotypes in generative text inference tasks. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4052–4065, Online. Association for Computational Linguistics.

Alexandra Uma, Tommaso Fornaciari, Anca Dumitrache, Tristan Miller, Jon Chamberlain, Barbara Plank, Edwin Simpson, and Massimo Poesio. 2021a. SemEval-2021 task 12: Learning with disagreements. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 338–347, Online. Association for Computational Linguistics.

Alexandra N Uma, Tommaso Fornaciari, Dirk Hovy, Silviu Paun, Barbara Plank, and Massimo Poesio. 2021b. Learning from disagreement: A survey. *Journal of Artificial Intelligence Research*, 72:1385–1470.

Ellery Wulczyn, Nithum Thain, and Lucas Dixon. 2017. Ex machina: Personal attacks seen at scale. In *Proceedings of the 26th International Conference on World Wide Web*, WWW '17, page 1391–1399, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

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

Xin Zhang, Guangwei Xu, Yueheng Sun, Meishan Zhang, and Pengjun Xie. 2021b. Crowdsourcing learning as domain adaptation: A case study on named entity recognition. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5558–5570, Online. Association for Computational Linguistics.

A Appendix

A.1 Baseline Results on Accuracy, Macro F1, TDV, JS Divergence

Building on the results in §5.1, we further investigate the effect of the level of disagreement on the passive and active learner baselines. In Figure 8, we compare these two baselines using both accuracy-based and distribution-based metrics.

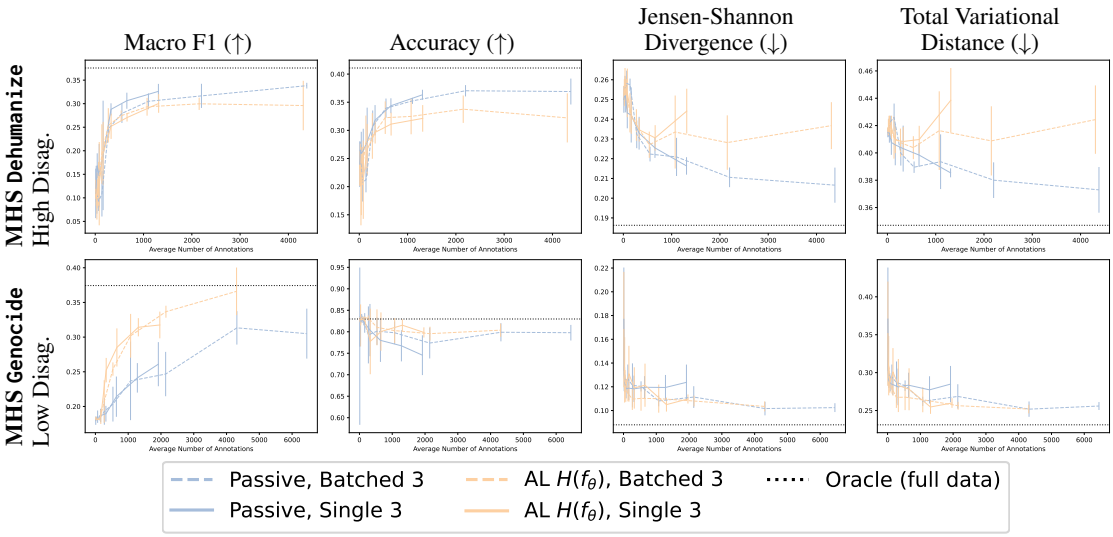
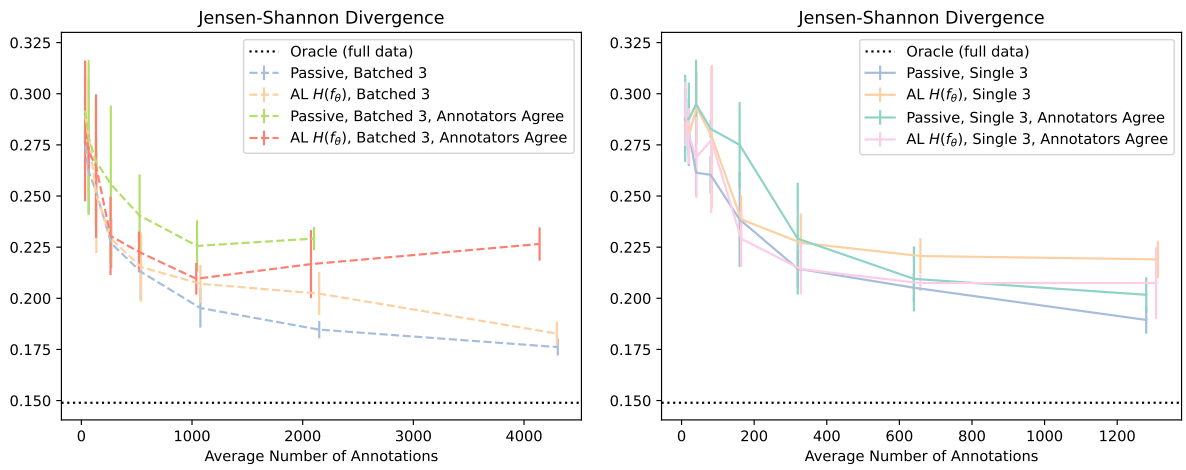


Figure 8: Comparison of passive and active leaner baselines on a high and low disagreement MHS attribute.

On the high disagreement attribute, `Dehumanize`, we see that passive learning still outperforms active learning when using accuracy-based measures, Macro F1 and Accuracy, though the effect is more subtle than with the distributions-based measures, JS Divergence and TVD.

For the low disagreement attribute, `Genocide`, we see that passive learning achieves the same performance as active learning in fewer annotations when considering Accuracy, JS Divergence, and TVD. For Macro F1, we see a much stronger trend, with the performance of the passive learner plateauing before the active learner. Noting how quickly all baselines achieved high accuracies, we argue that these trends are caused by the heavy class imbalance in the `Genocide` attribute which is heavily skewed to non-genocidal examples (See §A.5).



(a) Batched

(b) Single

Figure 9: Standard training vs training on only examples with full annotator agreement on MHS `Respect`.

To more directly investigate the effect of the level of disagreement on baseline model performance, we consider alternative train sets containing only examples with full annotator agreement. In other words, we use a subset of the original unlabeled data where all N available annotations have the same label value y.

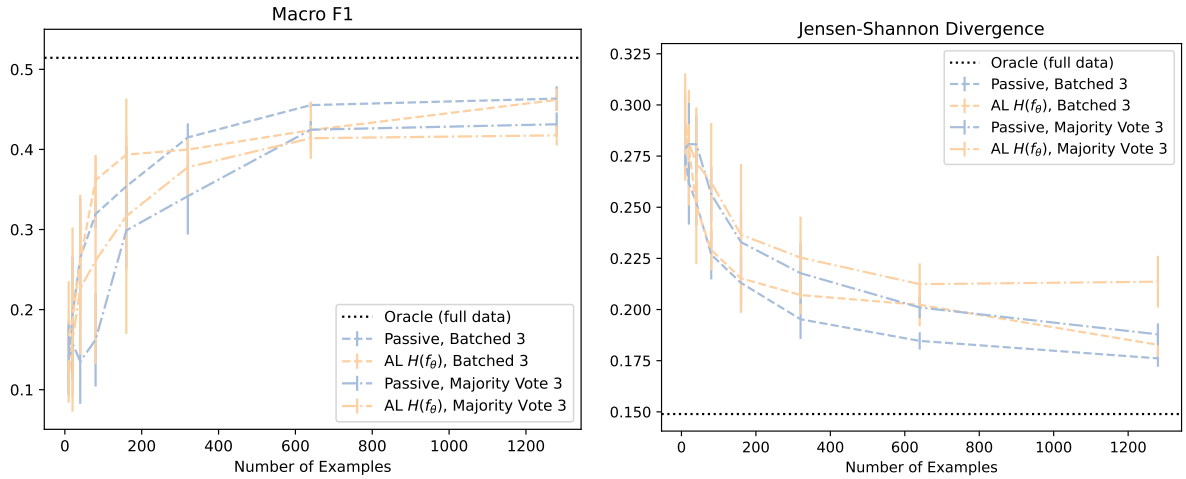
When querying for all available annotations (Figure 9a), the passive learner outperforms the active learner when they have access to the full training set. When they can only access training examples with full annotator agreement, the relationship is reversed.

When querying for single annotations at a time (Figure 9b), we still find that the passive learner performs better on the full training set. Using the training set with full annotator agreement, the active learner performs better earlier in training, but the final performance is not significantly different.

These results further show that model entropy alone isn't a good metric when humans disagree, which leads the passive approach, which simply picks at random, to perform better than the active learner.

A.2 Majority Vote

As we discussed in §3.1, we choose to use soft labels over majority vote labels which obscure disagreement. We compare training on majority votes to training directly on crowd annotations by treating each annotation as a separate learning instance (Uma et al., 2021b) for both passive learning and simple entropy-based active learning.



(a) (b)

Figure 10: Comparison of training on hard labels via majority vote vs soft labels with N annotations on MHS `Respect`

For both metrics distribution-based and accuracy-based metrics, we see a significant disadvantage when using hard labels. Considering Macro F1 (Figure 10a), using majority votes decreases the performance of the passive and active learners by 7.43% and 10.6% respectively. Considering Jensen-Shannon Divergence (Figure 10b), using majority votes decreases the performances by 6.25% and 14.4% respectively.

For both metrics, we see that by the end of training, using soft vs hard labels, not the querying method, determines which methods will be most successful. We see that the active batched model (weaker than its passive counterpart) does as good or better than the passive majority vote model. This confirms that aggregating annotation by majority vote can hurt performance when annotators disagree.

A.3 DAAL Improvements on Accuracy, Macro F1, TDV, JS Divergence

In this section, we show the full graphs of the JS Divergence results listed in Table 2 as well as for accuracy, macro F1, and total variational distance.

In Figure 11, we compare to the active learning baselines. For the MHS datasets, this tended to be the weaker baseline, with DAAL strongly outperforming both baselines on distribution-based metrics. Results on accuracy-based metrics were weaker on average, especially for `Genocide`. We see similar trends with `Toxicity-5`, though the JS Divergence is slightly worse on average at the optimal point.

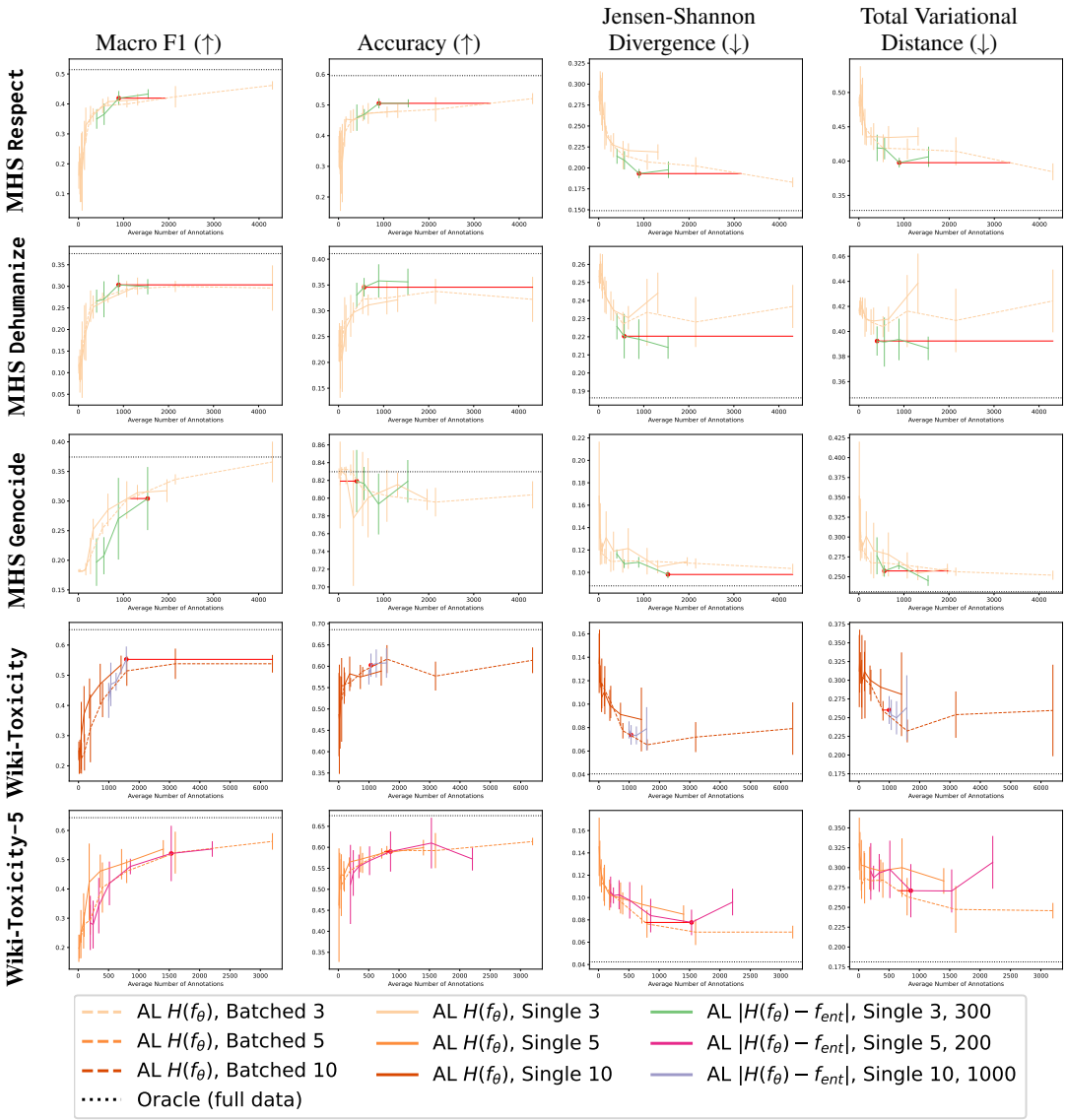


Figure 11: Comparison of DAAL (green, purple, or pink based on annotations per example) and entropy-based active learning (orange). The lines in red show DAAL's improvement in number of annotations. They connect the first measurement where DAAL was withing 5% of its best performance to the point where the batched active learning baseline achieves the same performance (if available).

In Figure 12, we compare to the passive learning baselines. The overall effects are similar to those in Figure 11. However, since the random baseline generally performed better than simple active learning in high disagreement settings (e.g., MHS `Dehumanize`), the improvements are generally weaker.

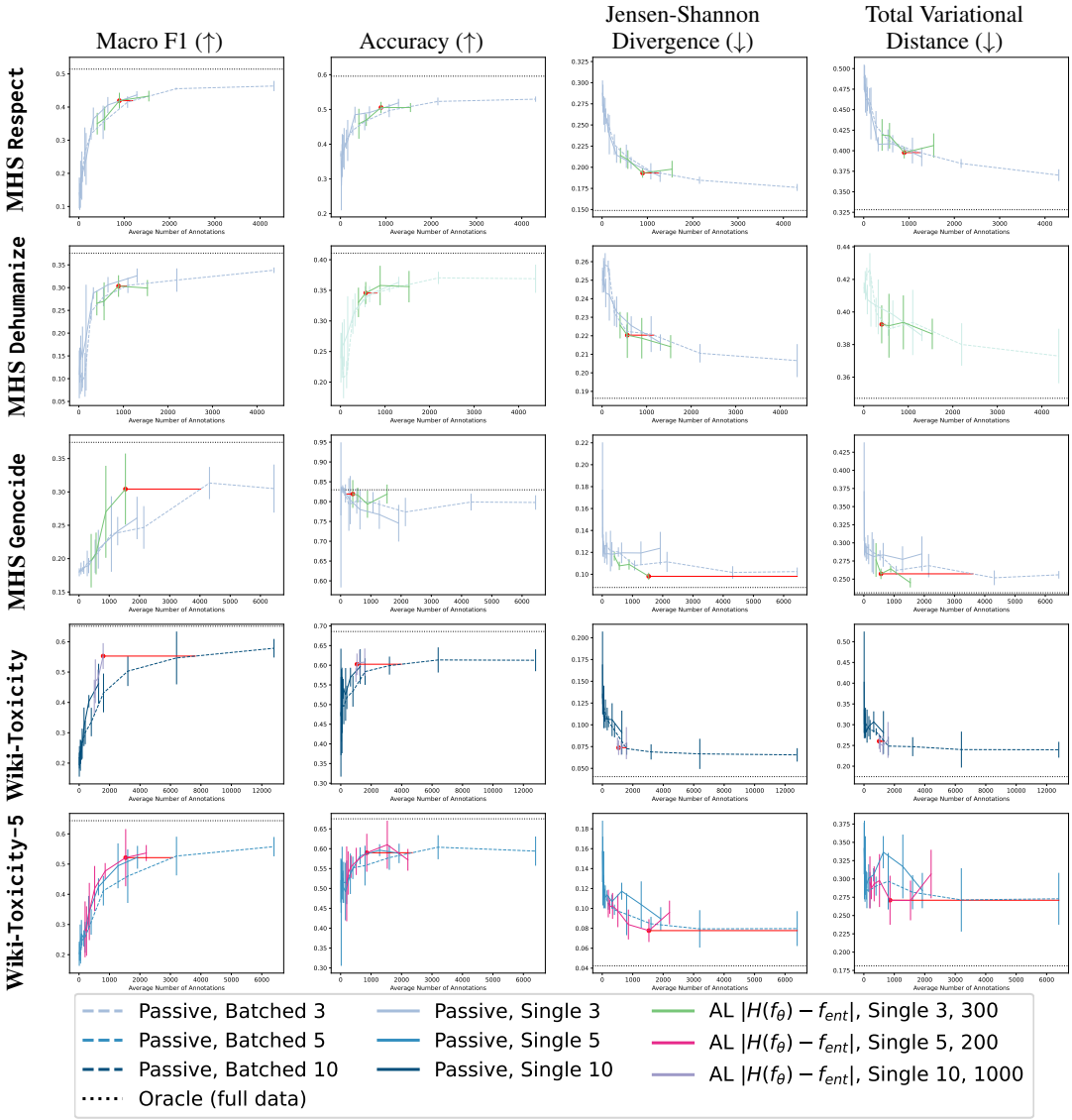
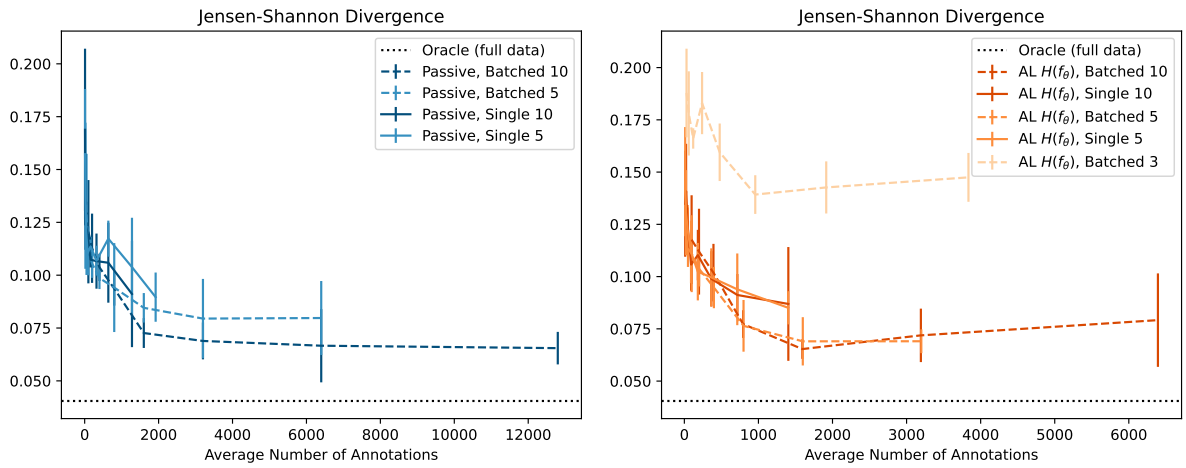


Figure 12: Comparison of DAAL (green, purple, or pink based on annotations per example) and passive learning (blue). The lines in red show DAAL's improvement in number of annotations. They connect the first measurement where DAAL was withing 5% of its best performance to the point where the batched passive learning baseline achieves the same performance (if available).

A.4 Annotations per Example

Here, we continue § 5.3's discussion of the effects of budget sizes and annotations per example. In Figure 5, we showed how the entropy predictor's performance on `Toxicity` does not significantly degrade until fewer than 5 annotations per example are available. In Figure 13, we can see that the 5 annotations passive learner sees a performance decrease. However, the baselines' overall performance did not drop significantly. On the other hand, in Figure 13b, we can see that the effect of decreasing to 3 annotations per example is much more significant.



(a) (b)

Figure 13: Baseline `Toxicity` results varying the number of annotations per example. We find that decreasing the annotations to 5 per example causes a small decrease in performance. Decreasing to 3 (a similar ammount to MHS) Significantly decreases the performance of the Batch AL model.

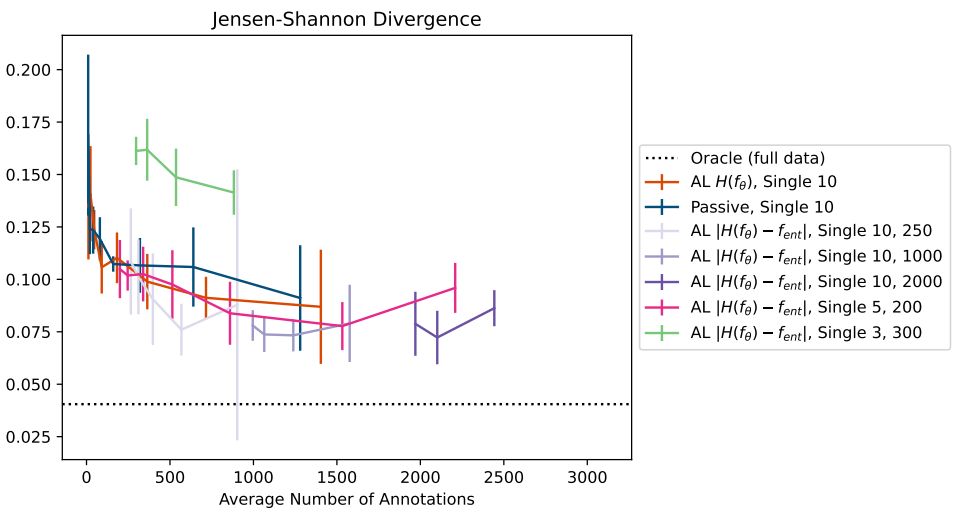
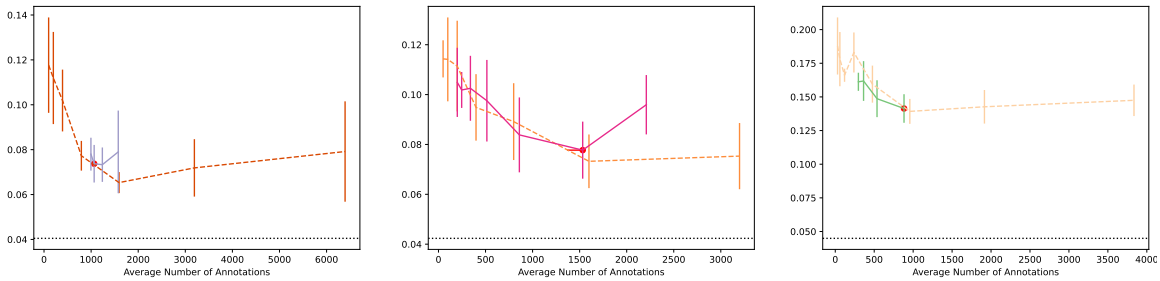


Figure 14: Comparison of performances on `Toxicity` when using different budgets for annotator entropy predictors described in the §3.5.

We find similar trends in DAAL when decreasing the number of annotations per example in 14. When we compare DAAL and entropy-based active learning using different numbers of annotations per example (Figure 15), we find a small trend of DAAL performing better in comparison to the baseline when the number of annotations per example is small, especially with as few annotations as MHS.

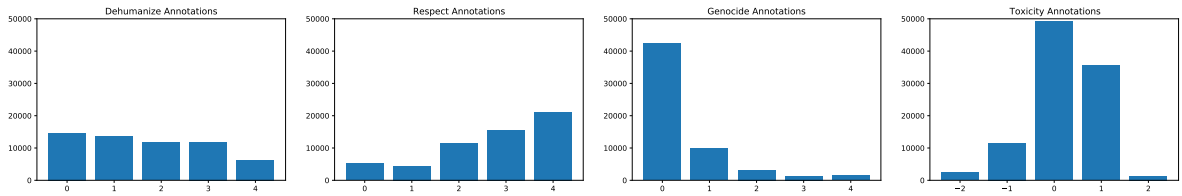


(a) 10 Annotations (b) 5 Annotations (c) 3 Annotations

Figure 15: DAAL vs AL $H(f_\theta)$ Single (orange) on varied annotations per example. On average DAAL can perform slightly worse than the baseline when the number of potential annotations is high.

A.5 Datasets' Vote Distributions

We show the vote distributions for the MHS dataset with `Respect`, `Dehumanize`, and `Genocide` attributes and the Wikipedia dataset with `Toxicity` attribute Figure 16.



(a) MHS `Dehumanize` (b) MHS `Respect` (c) MHS `Genocide` (d) Wikipeida `Toxicity`

Figure 16: Label distributions for MHS and Wikipedia `Toxicity` datasets.

Here, we have diverse settings. For instance, `Genocide` has the lowest level of disagreement between two random annotators (See Table 1), and we can see the majority of labels concentrate between two labels with the most examples of non-`Genocide` data. The `Respect` and `Toxicity` attributes have approximately the same level of disagreement with almost a 50% chance that two random annotators disagree. However, the distributions are quite different. The `Toxicity` label distribution has mostly two labels in use: neutral and toxic. This is similar to `Genocide` with the majority votes distributed between two labels: "strongly disagree" and "disagree" that text relates to genocide. The `Respect` attribute has annotations distributed between all labels, forming a left-skewed distribution, showing more different perspectives on this attribute. `Dehumanize` has the highest disagreement level. There is almost a 70% chance of two annotators disagreeing and the label distribution is almost uniform. This shows that there are enough examples that are seen differently by annotators (See Table 1).

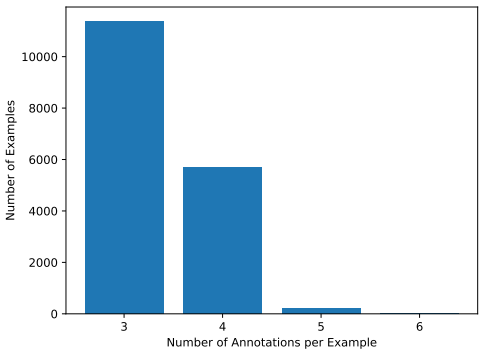


Figure 17: Annotations per example on our used portion of the MHS dataset. This excludes reference set examples (with > 200 annotations) and examples with less than 3 annotations.

The original MHS dataset contains both a reference set containing examples with more than 200 annotations per example and a larger set of examples with 1-6 annotations. As we discussed in §4.1, we use in our experiments a subset of the MHS dataset with 3-6 annotations (with an average of 3.35). The distribution of annotations per example in the data used in our experiments is shown in Figure 17.

A.6 Additional Experimental Details

For both our task and entropy prediction models, we use RoBERTa-Base models with 354 million parameters (Liu et al., 2020). They are trained using HuggingFace's transformers library.

The time it takes to train DAAL depends on the number of annotations per example, as each annotation is treated as a separate training instance. For the MHS dataset (average 3.35 annotations per example), it generally took < 15 hours to train DAAL on 1280 annotations. The bulk of this time is spent in inference, finding the task model's uncertainty on the ~ 15000 training examples. Our experiments were run on a single Intel Xeon E5405 GPU.

The two datasets used in our experiments, the MHS and Wikipedia Talk, are released under released under CC-by-4.0 and CC0 licenses respectively.