An Analysis of Active Learning Algorithms using Real-World Crowd-sourced Text Annotations

Varun Totakura
Department of Computer Science
Florida State University

Ankita Singh
Department of Computer Science
Florida State University

Yushun Dong
Department of Computer Science
Florida State University

Shayok Chakraborty
Department of Computer Science
Florida State University

***Abstract*—Active learning algorithms automatically identify the most informative samples from large amounts of unlabeled data and tremendously reduce human annotation effort in inducing a machine learning model. In a conventional active learning setup, the labeling oracles are assumed to be infallible, that is, they always provide correct answers (in terms of class labels) to the queried unlabeled instances, which cannot be guaranteed in real-world applications. To this end, a body of research has focused on the development of active learning algorithms in the presence of imperfect / noisy oracles. Existing research on active learning with noisy oracles typically simulate the oracles using machine learning models; however, real-world situations are much more challenging, and using ML models to simulate the annotation patterns may not appropriately capture the nuances of real-world annotation challenges. In this research, we first collect annotations of text samples (from 3 benchmark text classification datasets) from crowd-sourced workers through a crowd-sourcing platform. We then conduct extensive empirical studies of 8 commonly used active learning techniques (in conjunction with deep neural networks) using the obtained annotations. Our analyses sheds light on the performance of these techniques under real-world challenges, where annotators can provide incorrect labels, and can also refuse to provide labels. We hope this research will provide valuable insights that will be useful for the deployment of deep active learning systems in real-world applications. The obtained annotations can be accessed at https://github.com/varuntotakura/al_rcta/.**

***Index Terms*—Active Learning, Crowd-Source annotations, Text mining**

I. Introduction

The unprecedented growth of digital data in today's world has expanded the possibilities of solving a variety of real-world problems using computational learning frameworks. However, annotating data with class labels to induce a reliable deep neural network (DNN) model has remained a fundamental bottleneck. *Active Learning (AL)* algorithms alleviate this challenge by intelligently querying the labels of the most informative samples [1]. This drastically reduces the human annotation effort, as only a few unlabeled samples that are selected by the algorithm need to be labeled by the oracles [1]. AL has been extensively studied in text mining [2], computer vision [3], bioinformatics [4], anomaly detection [5] and a variety of other applications, with promising empirical results.

Conventional AL algorithms assume that the labeling oracles always provide the correct class labels to the queried unlabeled samples. This assumption may not hold for several real-world applications, where we usually have multiple labelers providing different qualities of annotation. For instance, in a crowdsourcing platform like the Amazon Mechanical Turk (AMT), hundreds of annotators are available who provide varying degrees of noisy annotations. This has motivated the development of AL algorithms which operate in the presence of imperfect / noisy annotators [6]–[8]. However, existing techniques typically simulate the imperfect annotators using classification models, such as random forests, SVMs etc. The models are trained on a subset of the training data and their prediction for a queried unlabeled sample is interpreted as the response of an imperfect oracle for that unlabeled sample [6], [9]. Another approach is to cluster the data into k clusters using the k-means algorithm, and to assume that each annotator is an expert on a particular cluster, where their labeling coincides with the ground truth; for samples in other clusters, they commit a fixed percentage of labeling errors [10]. While these methods offer a more practical solution to AL than the conventional assumption of noise-free annotators, using machine learning algorithms to model the imperfect annotators may not appropriately capture the nuances of real-world annotation challenges. Human annotation behavior is complex, and it may be a challenge to model it using simple machine learning techniques. Further, AL algorithms are most relevant in the training of data-hungry deep neural networks (DNNs), which have depicted commendable performance in a variety of real-world applications. This necessitates a thorough validation of AL algorithms, in conjunction with deep neural networks (deep active learning), using annotations collected from actual human oracles. In this paper, we attempt to address this relevant problem. Our contributions in this work can be summarized as follows:

- We collect crowd-sourced annotations for three bench-

This research was supported in part by the National Science Foundation under Grant Number: 2143424 (NSF CAREER Award)

[1]we use the terms oracles, labelers, annotators and users interchangeably in this paper

mark text classification datasets through a crowd-sourcing platform, where annotators can provide incorrect annotations, and can also refuse to provide annotations. The obtained annotations are made publicly available to foster further research on this important topic.
- We study the performance of 8 different AL techniques (conventional, as well as those designed to operate with imperfect oracles) with DNNs, using the collected annotations. Our analysis provides valuable insights into the performance of AL algorithms under real-world annotation constraints, which will hopefully be helpful for their deployment in practical applications.

II. Related Work

We present a brief survey of active learning algorithms in general, followed by a survey of AL in the presence of noisy oracles.

Active Learning: Active Learning (AL) is a well-researched problem in the machine learning literature [1]. With the popularity of data-hungry deep neural networks, researchers have studied the problem of deep active learning (DAL), where the goal is to automatically select the informative samples for manual annotation and simultaneously learn discriminating feature representations using a deep neural network [11]–[14]. Common DAL techniques include a task agnostic scheme which learns a loss prediction function to predict the loss value of an unlabeled sample and queries samples accordingly [3], a greedy technique to query a *Coreset* of samples that represent the whole dataset [15], a sampling technique based on diverse gradient embeddings (*BADGE*) [16], a strategy based on temporal output discrepancy that queries samples based on the discrepancy of outputs given by the models at different optimization steps during training [17] and an AL framework that queries unlabeled samples that can provide the most positive influence on model performance [18]. Methods based on adversarial training have particularly shown promising performance for deep AL [19]–[21].

AL with Imperfect Oracles: A body of research has focused on the development of AL algorithms in the presence of noisy annotators, where the goal is to select informative samples, as well as the best annotators for labeling them [22]. A few works have assumed a very simplistic setting, such as a single oracle which can provide incorrect labels and can also abstain from labeling [8], or two labeling oracles, one of which always returns the correct label and the other returns incorrect labels with a fixed probability [7], [23].

For multiple noisy annotators, a common approach is to use relabeling, where an actively queried sample is relabeled multiple times, and the final label is obtained using majority voting [24]–[26]. Zhao *et al.* [27] proposed an incremental relabeling strategy that identified the unlabeled instances that are most informative to label, and also samples that might benefit from relabeling, because their initial labels may be incorrect. The *ActiveLab* algorithm [28] is also based on a similar rationale, and balances between relabeling already labeled samples vs. labeling completely new samples. Another strategy to deal with multiple imperfect annotators is to estimate the single best oracle to annotate every queried sample. Yan *et al.* [10], [29] proposed a probabilistic multi-labeler model to compute the accuracy of each labeler and select the most confident labeler for each queried unlabeled sample. Huang *et al.* [6] proposed a cost effective active learning (*CEAL*) framework for active sample selection in the presence of multiple noisy oracles. Chakraborty [9] proposed an LP based formulation to identify the uncertain and diverse samples for annotation, and the optimal oracle to annotate each queried sample.

While these methods have depicted promising empirical results, the imperfect annotators in all these techniques were simulated using machine learning models, which may not be able to appropriately capture the complex nature of human annotation behavior. Yan *et al.* [10], [29] analyzed the performance of AL algorithms on scientific text data annotated by human experts. However, their experimental studies have the following limitations: (i) They only addressed binary classification problems, whereas real-world datasets typically involve multiple classes; studying the AL performance on binary problems may not appropriately reflect the challenges and complexities associated with multiple noisy annotators, who may provide incorrect class labels. (ii) They assumed that an annotator will always provide a label for a queried sample. Our collected annotations do not make this assumption, and annotators can even refuse to provide labels to queried instances, which is a very plausible scenario in real-world applications. (iii) These methods do not use deep learning. In today's deep learning era, a study of AL algorithms with noisy annotators, in conjunction with deep neural networks, is necessary, but lacking.

In this paper, we conduct a thorough analysis of the performance of different active learning algorithms, in conjunction with deep neural networks, with crowd-sourced annotations, on three multi-class textual datasets, where annotators can provide incorrect annotations and can even refuse to provide annotations. *To the best of our knowledge, this has not been studied in the literature*. We hope the insights gained from this research will enable AL to reach practical applications.

III. Datasets and Annotations

Datasets. We used three benchmark text classification datasets in this research: (i) **AG News** [30], which is a collection of news articles with 4 classes: *world, sports, business* and *science / technology*; (ii) **Consumer Complaints** [2], which contains complaints received about financial products and services and consists of 6 classes: *debt collection, prepaid card/debit card, mortgage, checking/savings account, student loan* and *vehicle loan/lease*; and (iii) **Wikipedia Movie Plots**[3], containing summary descriptions of movie plots scraped from Wikipedia, with 4 classes (movie genres): *drama, comedy, horror* and *action*.

Annotator Recruitment. We used the *Upwork* crowd-sourcing platform (https://www.upwork.com/) due to its wide

[2]https://catalog.data.gov/dataset/consumer-complaint-database

[3]https://www.kaggle.com/datasets/jrobischon/wikipedia-movie-plots

global reach and the flexibility it offers in managing project-based tasks. The job was made publicly available to eligible workers worldwide, with the condition that they be comfortable reading and understanding English, since the textual data and the instructions were all written in English. Consent was obtained from each worker to participate in the study. No information about the identity of the workers was recorded.

Annotation Protocol. We selected $3,000$ samples from each of the three datasets at random. The workers were provided with the list of possible classes for each dataset and were asked to annotate each sample with at most one label category. If they were unsure about the label of a sample and wanted to abstain from labeling, they were asked to enter 0 as the label. After accepting the job, each worker was given 30 days' time to submit their annotations. Once a job was submitted by a worker, we manually checked the file for errors, in particular, whether each sample was annotated with one of the valid class labels or 0. In case of any discrepancies, the worker was contacted to fix the issue. Upon a successful submission, each worker was provided with compensation for his / her efforts. Each set of $3,000$ samples was annotated by different crowd-sourced workers (10 for AG News and Consumer Complaints, and 9 for Wikipedia). The data collection protocol was reviewed and approved by an ethical review committee.

Analysis. Table I depicts an analysis of the obtained annotations. We note that the annotation accuracy varies quite significantly across different datasets. For the AG News and Consumer Complaints datasets, it was moderately high ($\approx 74\%$ and $\approx 78\%$ respectively), whereas for the Wikipedia dataset, it was much lower ($\approx 56\%$). Table I also demonstrates that the annotators refused to annotate some of the samples; however, this percentage is relatively low.

Dataset	Ann Acc (%)	Not Annotated (%)
AG News	74.13 ± 19.53	3.41 ± 4.27
Consumer Complaint	78.63 ± 10.06	7.90 ± 7.22
Wikipedia	56.50 ± 23.62	3.04 ± 4.03

TABLE I: Crowd-sourced annotation statistics for each dataset. The values are averaged across all the crowd-sourced annotators for each dataset.

We now study the performance of AL algorithms with the obtained annotations.

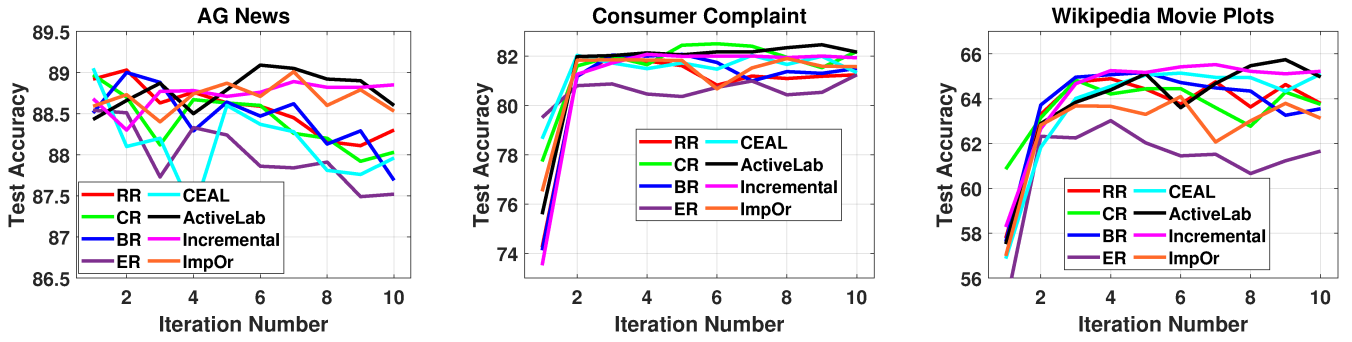
IV. Experiments and Results

Experimental Setup. Each dataset was divided into 3 parts: (i) an initial training set (500 samples) with clean annotations; (ii) an unlabeled set ($2,500$ samples); and (iii) a test set ($10,000$ samples) with clean annotations. A random subset of $2,500$ samples out of the $3,000$ samples whose annotations were obtained from crowd-sourcing, constituted the unlabeled set (the remaining 500 samples were used to compute the labeling accuracy of each oracle, detailed below). When a (sample-annotator) query was posed by an algorithm, the label provided to the corresponding sample, by the corresponding annotator was retrieved (instead of the ground truth label); the selected sample, together with the label was appended to the training set. If the retrieved label was 0 (the annotator refused to provide a label to this sample), the sample was simply discarded. After each AL iteration, the model was updated with the augmented training set and tested on the test set. The process was continued iteratively until a stopping condition was satisfied (taken as 10 iterations in this work). The objective was to study the improvement in performance on the test set with increasing sizes of the training set.

Each oracle was assigned an integer between 1 and 10 in increasing order of labeling accuracy, which was interpreted as the labeling cost C_i of oracle O_i for a given dataset, that is, the price to be paid to get one unlabeled sample from the dataset labeled by the particular oracle (more accurate oracles have higher costs). The labeling accuracy of each oracle was computed on the remaining 500 samples out of the $3,000$ samples (whose annotations were obtained from crowd-sourcing) by comparing the obtained annotations with the ground truth labels. Once a sample was annotated by a given oracle (be it correct or incorrect) the corresponding labeling cost of the oracle was deducted from the available query budget. The query budget B in each AL iteration was set as 100. All the results were averaged over 3 runs to rule out the effects of randomness.

Active Learning Baselines. We studied the performance of 8 AL algorithms in this research: (i) *Random*, where a batch of unlabeled samples was queried at random; (ii) *Entropy* [31] (iii) *Coreset* [15] (iv) *BADGE* [16] (v) *CEAL* [6] (vi) *ActiveLab* [28]; (vii) *Incremental* [27]; and $(viii)$ *AL with Imperfect Oracles (ImpOr)* [9]. The baselines *CEAL, ActiveLab, Incremental* and *ImpOr* are designed to operate with imperfect oracles; they select unlabeled samples, together with corresponding annotator(s) to annotate those samples. Among them, *ActiveLab* and *Incremental* use a relabeling strategy where the same sample can be labeled multiple times by different annotators, while *CEAL* and *ImpOr* select the single best optimal annotator for each queried sample. The other baselines (*Random, Entropy, Badge* and *Coreset*) merely select a batch of unlabeled samples; for these baselines, an oracle was selected at random to annotate the selected samples. They are thus denoted by Random-Random (*RR*), Entropy-Random (*ER*), Badge-Random (*BR*) and Coreset-Random (*CR*) respectively. These baselines were selected to capture the state-of-the-art methods in conventional AL, as well as the two most commonly used strategies in AL with noisy oracles: relabeling and optimal oracle selection. We used the BERT model [32] as the backbone DNN architecture in our experiments (for all the methods, for consistency) due to its popularity in text classification applications.

Implementation Details: The BERT model was trained using the conventional cross entropy loss; we used a learning rate of $1e^{-5}$ (with cosine annealing), momentum of 0.9, weight decay of $5e^{-7}$, and the *Adam* optimizer. The experiments were performed on a workstation with 64 GB RAM and



(a) AG News (b) Consumer Complaint (c) Wikipedia Movie Plots

Fig. 1: Study of AL performance. Methods that relabel a queried sample multiple times tend to produce better results than methods that label a queried sample using a single best annotator. No single method delivers the best performance consistently. Best viewed in color.

two NVIDIA Quadro RTX 5,000 GPUs with 16 GB memory each. The implementations were performed using PyTorch; the Python packages used include Hugging Face Transformers for BERT/Bert Tokenizer, Scikit-Learn, NumPy, Pandas, and Scipy.

	RR	ER	CR	BR	CEAL	ActiveLab	Incremental	ImpOr
RR								
ER	**0.00**							
CR	**0.02**	**0.00**						
BR	0.14	**0.02**	0.37					
CEAL	**0.01**	0.19	**0.04**	**0.04**				
ActiveLab	0.08	**0.00**	**0.02**	**0.02**	**0.00**			
Incremental	0.11	**0.00**	**0.03**	**0.05**	**0.00**	0.29		
ImpOr	0.13	**0.00**	**0.02**	**0.04**	**0.00**	0.15	0.30	

TABLE II: P-value matrix: AG News.

	RR	ER	CR	BR	CEAL	ActiveLab	Incremental	ImpOr
RR								
ER	0.47							
CR	**0.01**	**0.01**						
BR	**0.04**	0.36	**0.02**					
CEAL	**0.05**	**0.02**	0.18	0.12				
ActiveLab	**0.00**	0.06	0.36	**0.00**	0.39			
Incremental	**0.02**	0.27	0.11	0.10	0.25	**0.02**		
ImpOr	**0.03**	0.12	**0.02**	0.17	0.11	**0.04**	0.42	

TABLE III: P-value matrix: Consumer Complaint.

A. Study of Active Learning Performance

The active learning performance results are depicted in Figure 1. In each graph, the x-axis denotes the AL iteration number and the y-axis denotes the accuracy on the test set. We also conducted statistical tests of significance using *paired t-tests* to assess whether the difference in performance between two methods is statistically significant. Tables II, III and IV report the p-values between every pair of methods studied for the three datasets.

	RR	ER	CR	BR	CEAL	ActiveLab	Incremental	ImpOr
RR								
ER	**0.00**							
CR	0.43	**0.00**						
BR	0.29	**0.00**	0.45					
CEAL	0.37	**0.00**	0.47	0.48				
ActiveLab	0.19	**0.00**	0.36	0.36	0.29			
Incremental	**0.01**	**0.00**	0.10	**0.03**	**0.00**	**0.04**		
ImpOr	**0.00**	**0.00**	**0.01**	**0.00**	**0.01**	**0.00**	**0.00**	

TABLE IV: P-value matrix: Wikipedia Movie Plots.

AG News. *ActiveLab*, *Incremental* and *ImpOr* depict the best performance for this dataset. They, in general, outperform the methods that do not consider sample relabeling or optimal oracle selection (*RR, ER, CR* and *BR*); the improvement in performance over *ER, CR* and *BR* is statistically significant ($p < 0.05$). Among *ActiveLab, Incremental* and *ImpOr*, the difference in performance is not statistically significant ($p > 0.05$). *CEAL*, even though considers optimal oracle selection, does not perform well.

Consumer Complaint. *ActiveLab* outperforms *Incremental* and *ImpOr* at a significant level ($p < 0.05$). It also outperforms *CEAL* in the later AL iterations, although the improvement is not statistically significant. All the methods outperform *RR* at a significant level; they also outperform *BR*.

Wikipedia. *Incremental* depicts the best performance for this dataset; its improvement in performance is statistically significant for all the methods except *CR*. The methods that consider optimal oracle / sample relabeling (*ActiveLab, Incremental, CEAL*) tend to outperform the other methods, where the annotators are selected at random. *ImpOr*, however, depicts the worst performance, even though it considers optimal annotator selection.

Insights. We derive the following insights from our results:

- No single method delivers the best performance consistently across all the datasets, at a statistically significant level.

- AL algorithms designed to operate with imperfect annotators tend to outperform methods that assume the oracles to be infallible and focus on querying the informative samples only.
- Methods that relabel a queried sample multiple times using different annotators (*ActiveLab, Incremental*) tend to produce better results than methods that label a queried sample using only a single best annotator (*CEAL, ImpOr*). Thus, in a real-world setup with noisy annotators, it is probably a better strategy to annotate a given sample by multiple oracles and collate the annotations to derive the final label, rather than attempting to find the single optimal annotator for the sample.
- From Figure 1, we note that the AL curves for Consumer Complaint and Wikipedia depict an increasing trend, whereas those for AG News depict more or less a constant pattern. To study this further, we conducted experiments using the *ActiveLab* method with 50 and 100 training samples to start with (instead of 500, as in the original experiment). The results, shown in Figure 2, depict an increasing trend in accuracy. This corroborates that annotating samples beyond a certain point may not increase the accuracy (and may even adversely affect the accuracy). Thus, automatically deriving an appropriate stopping condition (when to stop annotating samples) is an important consideration in AL. While there has been some research in this area [33]–[35], it has not been explored in the context of noisy annotators. Conventionally, samples are annotated until the query budget is exhausted, which may not furnish the optimal performance always.

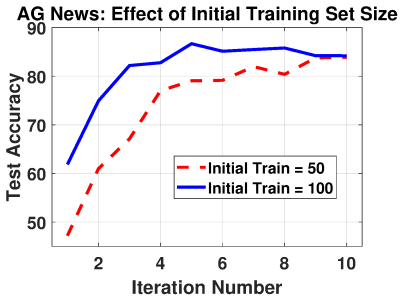


Fig. 2: Effect of initial training set size: *ActiveLab* on AG News dataset. Best viewed in color.

Dataset	All Incorrect	All Correct
AG News	89	272
Consumer Complaint	36	671
Wikipedia	16	19

TABLE V: Number of samples (out of $3,000$) that were incorrectly and correctly annotated by all the annotators, for each dataset.

- Our analysis further revealed that certain samples in each dataset were labeled incorrectly by all the annotators consistently. Table V reports the number of samples in each dataset that were annotated incorrectly and correctly by all the annotators. An example from the AG News dataset, with all incorrect annotations, is shown below:
"Currency traders, investors and strategists are more bearish on the dollar than at any time in the past 18 months, a Bloomberg News survey indicates."
This sample actually belongs to class *"World"*, but was labeled as *"Business"* by 9 annotators and *"Sports"* by one annotator. This is perhaps understandable, as the class definitions of *"World"* and *"Business"* may share some overlap, and it may be difficult to distinguish samples of one class from the other. Such samples can be extremely difficult to handle, as they are labeled not only incorrectly, but also consistently by most of the annotators; thus, even AL techniques that relabel a sample multiple times with different annotators will fail to derive the correct label for such instances. A possible solution may be to allow annotators to specify a *second-best* preference along with their top preference, when they annotate a sample, and develop an AL technique which considers *ordered class annotations*. There is very little research in this area [36], and it is an interesting direction for future AL research.

B. Study of Labeling Accuracy

The goal of this experiment was to study the labeling accuracy of each of the AL methods. Table VI shows the average number of samples queried in each AL iteration by each method, together with the average number of samples that were annotated correctly (in parentheses).

Insights. We derive the following insights from our results:

- Methods that use a random oracle (*RR, ER, BR, CR*) query ≈ 20 samples in each iteration. This is expected, as the query budget is 100 and the oracles have a cost of 1 to 10; thus, the expected cost of annotation for each sample is 5, which amounts to 20 queries in each AL iteration.
- *Incremental* queries each sample multiple times from different annotators; since a price needs to be paid for each query, it can query only a few samples in each AL iteration. However, due to the multiple annotations, it has a very high success rate. The other methods query more samples, but their success rate is much lower; thus, they end up polluting the training set with incorrectly labeled samples, which can potentially degrade their performance. For instance, for Wikipedia, *ImpOr* queries ≈ 27 samples in each run, out of which only ≈ 13 are correctly annotated; thus, it introduces about 14 incorrectly labeled samples in each iteration, which accounts for its poor performance (Figure 1c). *Incremental* queries only 4 samples, out of which 3 are correctly annotated and depicts much better performance. Thus, querying a small, but correctly annotated batch may sometimes

Method	AG News	Consumer Complaint	Wikipedia
RR	18.68 ± 2.81 (13.26 ± 3.42)	18.8 ± 2.16 (13.53 ± 2.74)	20.06 ± 1.74 (11.43 ± 2.12)
ER	18.20 ± 3.32 (12.10 ± 3.31)	19.20 ± 2.09 (12.90 ± 2.07)	20.30 ± 2.62 (10.12 ± 1.78)
CR	18.26 ± 2.04 (13.66 ± 2.28)	18.26 ± 2.04 (12.77 ± 3.28)	20.46 ± 1.92 (11.1 ± 2.24)
BR	18.6 ± 2.15 (13.23 ± 2.51)	19.26 ± 2.37 (14.03 ± 2.09)	20.5 ± 1.89 (11.66 ± 2.49)
CEAL	16.36 ± 2.72 (7.7 ± 2.45)	10.13 ± 0.44 (8.23 ± 1.28)	17.0 ± 0.0 (13.97 ± 1.85)
ActiveLab	18.43 ± 0.76 (6.56 ± 3.48)	19.02 ± 0.08 (8.03 ± 3.92)	23.33 ± 2.25 (10.16 ± 2.54)
Incremental	4.33 ± 0.53 (3.33 ± 0.53)	4.23 ± 0.49 (3.06 ± 0.68)	4.07 ± 0.25 (3.03 ± 0.73)
ImpOr	10.08 ± 0.46 (2.86 ± 1.82)	14.23 ± 1.09 (4.5 ± 1.65)	27.63 ± 3.76 (13.03 ± 3.37)

TABLE VI: Study of labeling accuracy. Average number of samples queried (average number of samples correctly annotated) in each AL iteration for each method studied.

Dataset	RR	ER	CR	BR	CEAL	ActiveLab	Incremental	ImpOr
AG News	4.23 ± 1.38	12.79 ± 3.21	35.62 ± 4.91	25.53 ± 3.09	1465.93 ± 83.12	106.59 ± 41.39	30.07 ± 9.26	1338.19 ± 45.26
Consumer Complaint	5.62 ± 3.02	17.57 ± 1.09	35.66 ± 5.23	28.08 ± 2.51	1304.86 ± 298.05	92.34 ± 31.20	32.86 ± 8.67	1390.40 ± 169.31
Wikipedia	2.27 ± 1.82	18.69 ± 1.27	37.06 ± 4.45	23.54 ± 3.01	809.49 ± 300.97	80.72 ± 24.54	27.23 ± 9.49	2248.11 ± 1891.68

TABLE VII: Computation time analysis. Average time required (in seconds) to query a batch of unlabeled samples (one AL iteration) for each method studied.

outperform querying larger batches with noisier annotations, particularly for datasets like Wikipedia, where the annotation accuracy is low (Table I).

C. Computation Time Analysis

Table VII reports the average time required to query a batch of unlabeled samples (one AL iteration) for each method. *RR* has the least computation time, as it performs random selection of samples and annotators and does not involve any computations. *CEAL* evaluates all possible (sample-annotator) pairs and selects the optimal pair iteratively, and thus has a high computation time. *ImpOr* solves an LP problem over a large matrix, which increases the computation time. *ActiveLab* and *Incremental* employ an uncertainty based active sampling criterion together with multiple random oracle assignments and thus have much lower computation time. *ER* involves an entropy computation for each unlabeled sample and also has a low computation time. *BADGE* performs a gradient embedding and a sampling computation, followed by *k-means++* for sample selection; *Coreset* solves a mixed integer programming problem for sample selection. These operations also require moderate computation time, as reflected in Table VII.

D. Study of Labeling Budget

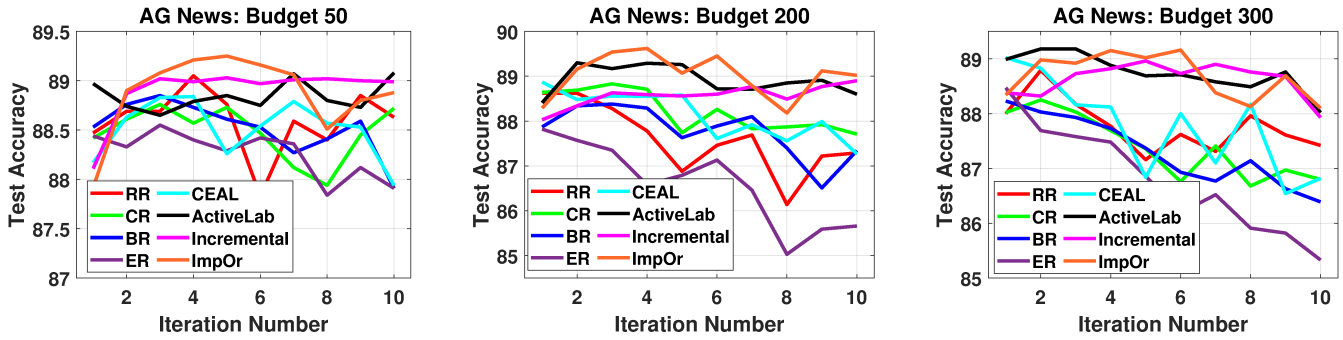
In this experiment, we studied the effect of the query budget per AL iteration. The results on the AG News dataset, for budgets $50, 200$ and 300, are depicted in Figure 3 (the budget for the results in Figure 1a was 100); the corresponding p-value matrices are depicted in Tables VIII, IX and X respectively. We notice a similar trend as in Figure 1a, where *ActiveLab, Incremental* and *ImpOr* outperform the other baselines. The improvement in performance achieved by these methods over *RR, ER, CR, BR* and *CEAL* is statistically significant across all the budgets $(p < 0.05)$. Among these three methods, there is no significant difference in performance for budgets 50 and 300, although *ActiveLab* and *ImpOr* outperform *Incremental* at a significant level for budget 200. These results reinforce our finding that AL methods designed to query informative samples in the presence of noisy annotators depict better performance than methods designed to query informative samples only. Further, AL methods which relabel a queried

	RR	ER	CR	BR	CEAL	ActiveLab	Incremental	ImpOr
RR								
ER	**0.01**							
CR	0.16	**0.02**						
BR	0.27	**0.00**	0.35					
CEAL	0.26	**0.01**	0.41	0.46				
ActiveLab	**0.03**	**0.00**	**0.01**	**0.02**	**0.01**			
Incremental	**0.01**	**0.01**	**0.01**	**0.01**	**0.00**	0.31		
ImpOr	**0.04**	**0.01**	**0.01**	**0.02**	**0.01**	0.41	0.37	

TABLE VIII: P-value matrix: AG News: Budget 50.

	RR	ER	CR	BR	CEAL	ActiveLab	Incremental	ImpOr
RR								
ER	**0.00**							
CR	**0.00**	**0.00**						
BR	0.19	**0.00**	**0.00**					
CEAL	**0.01**	**0.00**	0.27	**0.04**				
ActiveLab	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**			
Incremental	**0.00**	**0.00**	**0.05**	**0.00**	**0.04**	**0.01**		
ImpOr	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.21	**0.00**	

TABLE IX: P-value matrix: AG News: Budget 200



(a) Budget 50 (b) Budget 200 (c) Budget 300

Fig. 3: Study of labeling budget on the AG News dataset. *ActiveLab*, *Incremental*, and *ImpOr* outperform the other baselines across all budgets. Please refer to the text for more details. Best viewed in color.

	RR	ER	CR	BR	CEAL	ActiveLab	Incremental	ImpOr
RR								
ER	0.01							
CR	0.02	0.01						
BR	0.01	0.00	0.23					
CEAL	0.45	0.00	0.07	0.01				
ActiveLab	0.00	0.00	0.00	0.00	0.00			
Incremental	0.00	0.00	0.00	0.00	0.01	0.17		
ImpOr	0.00	0.00	0.00	0.00	0.01	0.29	0.31	

TABLE X: P-value matrix: AG News: Budget 300

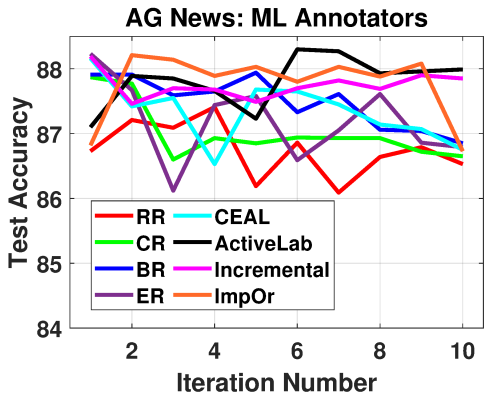


Fig. 4: Performance of AL algorithms using Machine Learning Models as Annotators on AG News dataset. Best viewed in color.

instance multiple times to ascertain the label (*ActiveLab* and *Incremental*) tend to outperform methods that attempt to ascertain the label based on the prediction from a single best annotator (such as *CEAL*). The *ER* method depicts poor performance (particularly, for budgets 200 and 300).

E. Performance using Machine Learning Models as Annotators

In this experiment, we study the performance of AL algorithms using machine learning (ML) models as annotators (rather than human annotators), as conventionally done in active learning research. We trained 10 machine learning models (*Random Forest, Gradient Boosting, AdaBoost, SVM, Logistic Regression, XGBoost, Decision Tree, CatBoost, KMeans, ExtraTrees*) on the initial training set; their accuracies ranged from 56.27% to 82.78%. When an unlabeled sample was queried, it was passed to the corresponding annotator (ML model) and the prediction of the model was used as the label of the unlabeled sample to retrain the underlying deep neural network.

The results on the AG News dataset are depicted in Figure 4. We note that using ML models as annotators, the accuracy of most of the methods remains more or less constant, or shows a decreasing trend. The accuracy values are also lower compared to the experiment when humans were used as annotators (Figure 1a). The accuracy of the *ImpOr* method, for instance, starts at ∼88.6% and exceeds 89% when humans are used as annotators (as evident from Figure 1a). With ML models as annotators, however, the accuracy starts at ∼87% and reaches a maximum value of 88.2%. The same applies for the *ActiveLab* method, whose accuracy also exceeds 89% with human annotators; but it attains the highest accuracy of 88.3% when ML models are used as annotators. This shows that using ML models as annotators can degrade the performance of AL algorithms; the results also corroborate the necessity of developing intelligent AL algorithms designed to work with imperfect human annotators, rather than relying on ML models as annotators.

V. Conclusion

The goal of this paper was to study the performance of active learning algorithms under real-world annotation challenges. To this end, we collected crowd-sourced annotations on 3 standard text classification datasets; we then conducted extensive experiments to study the performance of 8 AL algorithms using the collected annotations. To the best of our knowledge, this is the first research effort to study the performance of deep active learning algorithms, on multi-class

classification problems, with real-world crowd-sourced annotations. We hope this research will be a step toward bridging the gap between AL and real-world annotation challenges, and the insights gained will be useful for the deployment of AL systems in real-world applications.

REFERENCES

[1] B. Settles, "Active learning literature survey," in *Technical Report 1648, University of Wisconsin-Madison*, 2010.

[2] S. Tong and D. Koller, "Support vector machine active learning with applications to text classification," *Journal of Machine Learning Research (JMLR)*, vol. 2, pp. 45–66, 2001.

[3] D. Yoo and I. Kweon, "Learning loss for active learning," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[4] H. Osmanbeyoglu, J. Wehner, J. Carbonell, and M. Ganapathiraju, "Active machine learning for transmembrane helix prediction," *BMC Bioinformatics*, vol. 11, no. 1, 2010.

[5] T. Pimentel, M. Monteiro, A. Veloso, and N. Ziviani, "Deep active learning for anomaly detection," in *IEEE International Joint Conference on Neural Networks (IJCNN)*, 2020.

[6] S. Huang, J. Chen, X. Mu, and Z. Zhou, "Cost-effective active learning from diverse labelers," in *International Joint Conference on Artificial Intelligence (IJCAI)*, 2017.

[7] C. Zhang and K. Chaudhuri, "Active learning from weak and strong labelers," in *Neural Information Processing Systems (NIPS)*, 2015.

[8] S. Yan, K. Chaudhuri, and T. Javidi, "Active learning from imperfect labelers," in *Neural Information Processing Systems (NIPS)*, 2016.

[9] S. Chakraborty, "Asking the right questions to the right users: Active learning with imperfect oracles," in *AAAI Conference on Artificial Intelligence*, 2020.

[10] Y. Yan, G. Fung, R. Rosales, and J. Dy, "Active learning from crowds," in *International Conference on Machine Learning (ICML)*, 2011.

[11] P. Ren, Y. Xiao, X. Chang, P. Huang, Z. Li, B. Gupta, X. Chen, and X. Wang, "A survey of deep active learning," *ACM Computing Surveys*, vol. 54, no. 9, 2021.

[12] S. Khosla, C. Whye, J. Ash, C. Zhang, K. Kawaguchi, and A. Lamb, "Neural active learning on heteroskedastic distributions," in *arXiv:2211.00928v2*, 2023.

[13] S. Nuggehalli, J. Zhang, L. Jain, and R. Nowak, "Direct: Deep active learning under imbalance and label noise," in *arXiv:2312.09196v3*, 2024.

[14] N. Shafir, G. Hacohen, and D. Weinshall, "Active learning with a noisy annotator," in *arXiv:2504.04506v1*, 2025.

[15] O. Sener and S. Savarese, "Active learning for convolutional neural networks: A core-set approach," in *International Conference on Learning Representations (ICLR)*, 2018.

[16] J. Ash, C. Zhang, A. Krishnamurthy, J. Langford, and A. Agarwal, "Deep batch active learning by diverse, uncertain gradient lower bounds," in *International Conference on Learning Representations (ICLR)*, 2020.

[17] S. Huang, T. Wang, H. Xiong, J. Huan, and D. Dou, "Semi-supervised active learning with temporal output discrepancy," in *IEEE International Conference on Computer Vision (ICCV)*, 2021.

[18] Z. Liu, H. Ding, H. Zhong, W. Li, J. Dai, and C. He, "Influence selection for active learning," in *IEEE International Conference on Computer Vision (ICCV)*, 2021.

[19] S. Sinha, S. Ebrahimi, and T. Darrell, "Variational adversarial active learning," in *IEEE International Conference on Computer Vision (ICCV)*, 2019.

[20] J. Zhu and J. Bento, "Generative adversarial active learning," in *arXiv:1702.07956*, 2017.

[21] M. Ducoffe and F. Precioso, "Adversarial active learning for deep networks: a margin based approach," in *International Conference on Machine Learning (ICML)*, 2018.

[22] Y. Chen, K. Sankararaman, A. Lazaric, M. Pirotta, D. Karamshuk, Q. Wang, K. Mandyam, S. Wang, and H. Fang, "Improved adaptive algorithm for scalable active learning with weak labeler," in *arXiv:2211.02233v1*, 2022.

[23] P. Donmez and J. Carbonell, "Proactive learning: cost-sensitive active learning with multiple imperfect oracles," in *ACM Conference on Information and Knowledge Management (CIKM)*, 2008.

[24] P. Donmez, J. Carbonell, and J. Schneider, "Efficiently learning the accuracy of labeling sources for selective sampling," in *ACM Conference on Knowledge Discovery and Data Mining (KDD)*, 2009.

[25] Y. Zheng, S. Scott, and K. Deng, "Active learning from multiple noisy labelers with varied costs," in *IEEE International Conference on Data Mining (ICDM)*, 2010.

[26] P. Ipeirotis, F. Provost, V. Sheng, and J. Wang, "Repeated labeling using multiple noisy labelers," *Data Mining and Knowledge Discovery*, vol. 28, 2014.

[27] L. Zhao, G. Sukthankar, and R. Sukthankar, "Incremental relabeling for active learning with noisy crowdsourced annotations," in *International Conference on Social Computing*, 2011.

[28] H. Goh and J. Mueller, "Activelab: Active learning with re-labeling by multiple annotators," in *International Conference on Learning Representations Workshop (ICLR-W)*, 2023.

[29] Y. Yan, R. Rosales, G. Fung, F. Farooq, B. Rao, and J. Dy, "Active learning from multiple knowledge sources," in *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2012.

[30] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in *Neural Information Processing Systems (NeurIPS)*, 2015.

[31] Q. A. Wang, "Probability distribution and entropy as a measure of uncertainty," *Journal of Physics A: Mathematical and Theoretical*, vol. 41, 2008.

[32] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Nations of the Americas Chapter of the Association for Computational Linguistics (NAACL)*, 2019.

[33] Y. Zhang, W. Cai, W. Wang, and Y. Zhang, "Stopping criterion for active learning with model stability," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 9, 2017.

[34] J. Zhu, H. Wang, and E. Hovy, "Learning a stopping criterion for active learning for word sense disambiguation and text classification," in *International Joint Conference on Natural Language Processing (IJNLP)*, 2008.

[35] H. Ishibashi and H. Hino, "Stopping criterion for active learning based on deterministic generalization bounds," in *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2020.

[36] Y. Xue and M. Hauskrecht, "Active learning of multi-class classification models from ordered class sets," in *AAAI Conference on Artificial Intelligence*, 2019.