Real-Time Visual Feedback for Educative Benchmark Creation : A Human-and-Metric-in-the-Loop Workflow

Anjana Arunkumar aarunku5@asu.edu
Swaroop Mishra srmishr1@asu.edu
Bhavdeep Sachdeva bssachde@asu.edu
Chitta Baral chitta@asu.edu
Chris Bryan cbryan16@asu.edu

School of Computing, Informatics, and Decision Systems Engineering
Arizona State University

Abstract

Recent research has shown that language models exploit 'artifacts' in benchmarks to solve tasks, rather than truly learning them. Considering that this behavior inflates model performance, shouldn't the creation of *better benchmarks* be our priority? In pursuit of this, we focus on *guiding crowdworkers*, an under-explored facet of addressing benchmark idiosyncrasies. We propose VAIDA, a novel benchmark creation paradigm for NLP. VAIDA provides realtime visual feedback to both crowdworkers and backend analysts on both sample and dataset quality, and aims to educate them on the same. VAIDA also facilitates sample correction to improve quality via recommendations. VAIDA is domain, model, task, and metric agnostic, and constitutes a paradigm shift for robust, validated, and dynamic benchmark creation via human-and-metric-in-the-loop workflows. We demonstrate VAIDA's effectiveness by leveraging a state-of-the-art data quality metric DQI over four datasets. We further evaluate via expert review and a user study with NASA TLX. We find that VAIDA decreases effort, frustration, mental, and temporal demand of crowdworkers and analysts, while simultaneously increasing the performance of both user groups.

1 Introduction

Researchers invest significant effort to create benchmarks in AI, including ImageNet [4], SQUAD [18], and SNLI [1], as well as to create, tune, and tweak models that solve these benchmarks. *Can we rely on these benchmarks?* A growing body of recent research [22, 16, 11] is revealing that models exploit spurious bias– unintended correlations between input and output [23] (e.g. the word 'not' is associated with the label 'contradiction' in Natural Language Inference (NLI) [6])– instead of the actual underlying features, to solve many popular benchmarks. Models therefore fail to generalize, and experience drastic performance drops when testing with out of distribution (OOD) data or adversarial examples [2, 14, 26]. These biases[1] have led to the overestimation of AI's true advancement [21], and limit its deployment in safety-critical domains [8]. This begs the question: *Shouldn't ML researchers consequently focus on creating 'better' datasets rather than developing increasingly complex models on bias-laden benchmarks?*

Benchmark creators report bias baselines– hypothesis-only baseline in NLI [5])– and if the baseline

[1] Henceforth 'bias' implies spurious bias and also artifacts

Preprint. Under review.

performance is high, they might have to delete all the data created or can leverage adversarial filtering algorithms like AFLite [21] to delete targeted subsets of the data. This, along with other bias mitigation approaches [3, 10] has the following limitations: (i) data deletion/augmentation and residual learning do not justify the original investment in data creation, and (ii) crowdworkers are not provided continuous feedback to learn what constitutes high quality data– and so have additional overhead due to the manual effort involved in sample creation/validation. One potential solution to these problems is *in situ* feedback about artifacts while benchmark data is being created. *To our knowledge, there are no approaches which provide realtime artifact identification, feedback, and reconciliation opportunities to data creators, nor guide them on data quality*.

Contributions: (i) We propose *VAIDA* (Visual Analytics for Interactively Discouraging Artifacts), a novel system for benchmark creation that provides continuous visual feedback to data creators as the benchmark is being created. VAIDA supports both artifact identification and resolution, implicitly educating two classes of users – *crowdworkers* and *analysts*– on data quality. (ii) We design a *crowdworker* workflow and interface to create and submit new data samples for benchmark inclusion. Feedback from VAIDA guides crowdworkers on why a sample likely constitutes an artifact. To assist with sample modification, we propose an AutoFix module, that allows for machine-assisted sample modification to achieve higher quality (i.e., lower bias and higher generalizability). (iii) We develop a series of visualizations for *analysts* to review and verify submitted samples, as well as analyze and resample train-test splits to build an optimal dataset. VAIDA allows visual exploration of the effect of a sample's addition to a dataset in both cold-start and pre-existing data scenarios. We also propose the use of TextFooler for adversarial transformation to increase benchmark robustness using model-in-the-loop. (iv) We leverage DQI [15], a data quality metric that identifies artifacts by decomposing samples according to their language properties, as part of a metric-in-the-loop approach to demonstrate VAIDA's effectiveness over a set of four benchmarks. (v) We further evaluate VAIDA empirically through expert review and a user study to understand the cognitive workload it imposes. The results[2] indicate that VAIDA decreases mental demand, temporal demand, effort, and frustration of crowdworkers (29.7%) and analysts(12.1%); it increases performance by 30.8% and 26% respectively, and also educates crowdworkers on how to create *high quality* samples. VAIDA represents a novel, and to our knowledge, substantial shift in how benchmarks can be developed and validated, as it enables dynamic identification and resolution of artifacts during benchmark creation. By allowing crowdworkers and analysts to intuitively work in sync, spurious bias can be minimized, in turn reducing the overestimation of AI systems' capabilities; this enables their deployment in safety-critical domains.

2 Task Selection and Controlled Dataset Creation

In this work, we apply VAIDA for a natural language inference task (though it is task-independent), and mimic the SNLI dataset creation and validation processes. Elicited annotation has been found to lead to social bias in SNLI using probablistic mutual information (PMI) [20]. Visual feedback is provided based on DQI (which takes PMI into account) to explicitly correct this bias, and discourage the creation of such samples. Also, human annotation of machine-generated sentences/sentences pulled from existing texts instead of elicitation has been suggested to reduce such bias [25]. However, machine-generated text might look artificial, and work has shown that text generation has its own set of quality issues [13]. While we use AutoFix and TextFooler as modules to automatically transform samples, they are designed to be used in parallel with human sample creation. Their results can also be further modified by humans prior to submission. We see less reliance on these tools over the course of our user study (Subsection 5.3).

Additionally, previous work [19] in controlled dataset creation trains crowdworkers, and selects a subset of the best-performing crowdworkers for actual corpus creation. Each crowdworker's work is reviewed by another crowdworker, who acts as an analyst (as per our framework) of their samples. However, in real-world dataset creation, such training and selection phases might not be possible. Additionally, the absence of a metric-in-the-loop basis for feedback provided during training can potentially bias (through trainers) the created samples.

3 Workflow and Modules

In this section, we describe VAIDA's high-level workflow and important backend processes.

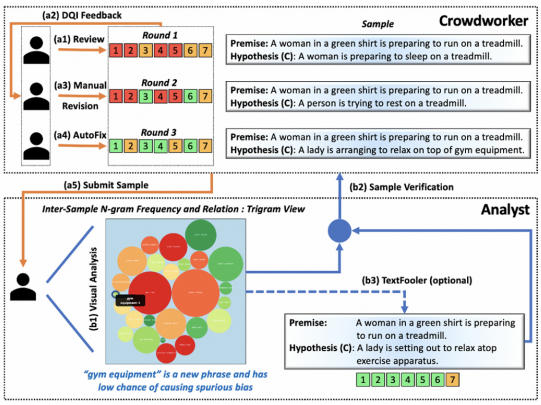
[2] Henceforth, red: decrease, and green: increase

3.1 Crowdworker and Analyst Workflows

VAIDA's high-level workflow is shown in Figure 1(A). Both crowdworkers and analysts work in parallel to create benchmark data points.

For crowdworkers, (a1) newly created samples are evaluated by DQI and (a2) realtime feedback is given to the user about potential biases. To fix an artifact, users can (a3) manually revise the sample, (a4) run AutoFix to automatically update it, or simply discard the sample and create a new one. After review (and potentially iterative DQI evaluations/revisions), (a5) the sample can be submitted for benchmark inclusion.

For analysts, (b1) VAIDA provides several visual interfaces to support detailed analysis and review of submitted samples, and to assess overall benchmark quality. Submitted samples enter a *pending* state until reviewed by the analyst, who *accepts*, *rejects*, or *modifies* the sample. (b2) Sample decisions are communicated back to crowdworkers to provide continuous feedback about performance and allow them to correct such samples. (b3) Analysts also have the option to submit low quality samples to TextFooler for adversarial transformation and augument with high quality samples to improve robustness of dataset, thereby ensuring minimal data loss.



S.No.	Component Name	DQI Implication	VAIDA Usage	Round 1	Round 2	Round 3	TextFooler
1	Vocabulary	Ambiguity and diversity of a dataset's language.	Does the sample contribute new words?	0 new words	0 new words	2 new words	3 new words
2	Inter-Sample N-gram Frequency and Relation	Word/phrase repetition and similarity between samples.	Does the sample contribute new combinations of words and phrases?	0 new phrases	0 new phrases	1 new phrases	2 new phrases
3	Inter-Sample STS	Syntactic, semantic, and pragmatic sentence parsing.	How similar is the hypothesis to all other premises or hypotheses?	2 cases highly similar	1 case highly similar	1 case highly similar	0 cases highly similar
4	Intra-sample Word Similarity	Word overlap and similarity within sample statements.	How similar are all words within the sample?	High Overlap	High Overlap	Low Overlap	Low Overlap
5	Intra-sample STS	Phrase/sentence level overlap similarity within a sample.	How similar is the hypothesis to the premise?	High Similarity	High Similarity	Moderate Similarity	Low Similarity
6	N-gram Frequency per Label	Distribution of samples according to annotation.	Is the hypothesis too obvious for the system?	sleep	rest	---	---
7	Inter-Split STS	Optimal dissimilarity between train and test samples.	Is the sample too similar to an existing sample?	1 case highly similar	1 case highly similar	1 case highly similar	1 case moderately similar

(B)

Figure 1: **(A)** VAIDA workflow– branches *(a)*: crowdworker, *(b)*: analyst functions. **(B)** Language properties considered in DQI, interpretation in VAIDA, and statistics for each feedback shown in **(A)** given 100 pre-existing dataset samples; STS: semantic textual similarity. C:"contradiction".

3.2 Modules

DQI and Traffic Signal Scheme: VAIDA communicates sample quality using an intuitive traffic signal color coding (red, yellow, green) to indicate if samples might lead to bias. The quality of individual features (aspects) of samples are evaluated based on decreasing presence of artifacts and increasing generalization capability. Based on overall sample quality, VAIDA computes the probability the sample will be accepted/rejected.

To demonstrate this, we leverage DQI [15], which can: (i) compute the overall data quality for a benchmark with n data samples, and (ii) compute the impact of a new $(n+1)^{th}$ data sample. When a crowdworker creates a new sample, DQI estimates its quality by calculating seven component values corresponding to a set of seven language properties; these are defined in Figure 1(B), along with their interpretation in VAIDA[3].

AutoFix: We propose AutoFix as a module to help crowdworkers avoid creating bad samples by recommending changes to a sample to improve its quality. The AutoFix algorithm is explained in Figure 2(a). Given a premise, hypothesis, and the DQI values for the hypothesis, AutoFix sequentially masks each word in the hypothesis and ranks words based on their impact on model output, i.e. their importance. DQI bins values into three classes: *high*, *acceptable*, and *low* quality. Hypothesis words are replaced in the order of importance to achieve *acceptable* quality. DQI hence controls the amount and aspect of changes made by AutoFix.

VAIDA employs AutoFix in an incremental manner to facilitate human-in-the-loop continuous feedback. By incrementally changing the sample one word at a time, users can understand how and why their sample is being modified and how DQI values are affected. Figure 1(A) shows AutoFix

[3]Hyperparameters depend on the application type. See Supplemental Material:Hyperparameters.

results after being used three times on a data sample, while Table 1 shows examples of AutoFix being applied to SNLI samples.

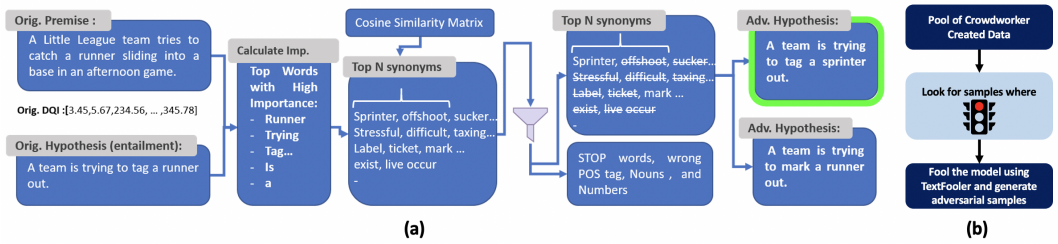


Figure 2: (a) AutoFix Algorithm. (b) TextFooler Algorithm.

Module	Premise	Orig. Hypothesis	DQI	Suggested Words	New Hypothesis	New DQI	Label
AutoFix	A man in a green apron smiles behind a food stand	A man smiles	3.324	smiles	A person is grinning	6.304	Entailment
TextFooler	One blond girl offers another blond girl some food	One girl offering another girl food	2.378	N/A	One girl tenders another food.	3.116	Entailment

Table 1: Examples for Autofix and TextFooler, with DQI's Intra-sample STS values for SNLI samples.

TextFooler: From an analyst's perspective, the quality of a submitted sample might be "too low" because (i) the crowdworker might not employ AutoFix, or (ii) there is a narrow acceptability range due to the criticality of the application domain, such as in BioNLP [12]. We therefore implement a module TextFooler [9] for adversarial sample transformation of low quality samples (instead of discarding them) to improve benchmark robustness (another key aspect in benchmark construction), and ensure that the crowdsourcing effort is not wasted.
We initially use AFLite, a recent adversarial filtering approach [2], to bin samples into *good* (retained samples) and *bad* (filtered samples) splits. Using TextFooler, we adversarially transform bad split data to flip the label; we revert back to the original label and evaluate this sample using DQI, as shown in Figure 1(A) and Table 1.

4 Interface Design Choices

VAIDA provides customized interfaces for both crowdworkers and analysts.

4.1 Crowdworker Interface

In addition to workflow functionalities, the crowdworker interface (Figure 3(a)) provides interface navigation, data creation, and feedback interpretation instructions (A). Sample creation (B) mimics the original SNLI crowdsourcing interface– examples (b1) are given, and the premise field (b2) autopopulates with captions from the Flickr30 corpus; three hypotheses (for entailment, neutral, contradiction labels) are to be entered at a time, though they are reviewed individually. DQI feedback (C) is shown for each component (c1), and hovering on these displays a tooltip that suggests sample fixes to improve quality (c2). (c3) Overall sample quality and (c4) estimated probability it will be accepted provide additional feedback. AutoFix (b3) can be used for automatic fixes. Samples enter a pending state (d1), until analyst review, upon which the count (d2) and pie chart (d3) update. Historical quality of samples submitted by the user (e1) , and (e2) current rank of the user are shown to help crowdworkers gauge their performance. Communication links for FAQs, and error reporting are also provided (F).

4.2 Analyst Interfaces

While crowdworkers work within a single tightly-coordinated interface to create, submit, and review samples, analysts can navigate between a set of nine interfaces (Figure 3(b)) to review samples in detail to make accept, reject, and modification decisions, and to assess overall benchmark quality.
(UI) The *single crowdworker view* provides a view similar to the crowdworker interface, and allows the analyst to review the work of a single crowdworker. The data creation panel is modified to allow the analyst to iterate over and review submitted samples. For low quality samples, the TextFooler module can be invoked (via a 'Generate Adversarial Example' button). (C1–C7) Other interfaces available to the analyst support detailed review of specific DQI components and allow the analyst to simulate how adding one or more submitted samples affects the benchmark's quality. Several visualization techniques are employed (treemap, node-link diagram, bubble chart, heatmap, bar chart,

etc.) tailored to the specific DQI component of interest, but all interfaces consistently utilize the traffic signal color scheme to represent quality[4].

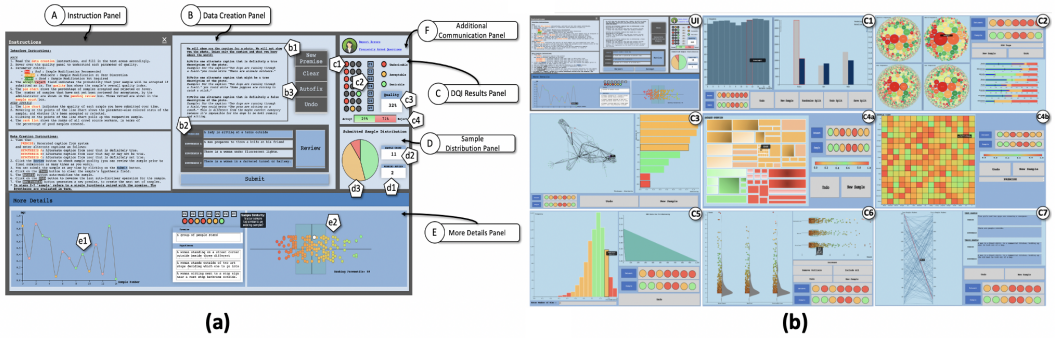


Figure 3: **(a)** VAIDA's crowdworker interface consists of six linked panels: (A) Instructions, (B) Data creation, (C) DQI results, (D) Sample distribution, (E) More details, and (F) Additional communication. **(b)** VAIDA's provides a collection of interfaces for the analyst supporting detailed analysis, review, and investigation of submitted samples and the overall state of the benchmark.

5 Evaluation

We evaluate VAIDA's efficacy at providing real time feedback to crowdworkers during benchmark creation using case studies, expert reviews, and user studies. First, we evaluate by assuming gold scores corresponding to the results for a recent adversarial filtering approach, AFLite [2]. Since this assumption has potential risks, as AFLite may have some limitations (outlined in [15]), expert review and the subsequent user study provide additional evaluation and feedback.

5.1 Case Studies

We evaluate our traffic signal scheme (based on DQI) over four datasets: SNLI [1], MNLI[24], SQUAD 2.0 [17], and Story CLOZE Task [22]. In the case of SQUAD 2.0 and Story CLOZE, we split each sample into multiple samples– for e.g., in Story CLOZE there are two ending choices per sample and so we make two samples, with label *True* for the sample with the correct ending and *False* for the sample with the incorrect ending.

The presence of a large number of artifacts has been shown in several studies on SNLI [6] and Story CLOZE Task [22]. MNLI and SQUAD 2.0 have been shown to have a relatively smaller number of artifacts [6, 11], and therefore ensure adversarial evaluation of VAIDA. We evaluate each dataset using its test sets, or if unavailable, on its dev sets.

Setup: For each dataset, we filter using AFLite and divide it into two categories: *good* and *bad*, where each category respectively refers to the set of samples retained and removed after adversarial filtering. Evaluation with VAIDA involves providing feedback in two different settings: (i) no preexisting samples, and (ii) 100 preexisting samples corresponding to the good category. For (ii), random sampling of 100 pre-existing samples is done 10 times, for a fair comparison.

In (ii), we: (a) compute DQI for the existing sample set as x_1, (b) recompute DQI for the sample set after a new sample is added as x_2, and (c) calculate $\Delta x = x_1 - x_2$. The crowdworker interface shows the DQI components corresponding to Δx. In the analyst interface, both Δx as 'sample' and x_2 as the 'dataset' quality are shown component-wise in each view. For fair comparison, we have taken illustrative samples from the AFLite paper [2] for SNLI. We randomly sample for other datasets[5], as corresponding examples were not illustrated in those papers.

Configuring the Boundary Separating Red, Yellow and Green Flags There exist two hyperparameters separating the boundary between red, yellow, and green flags. We tune hyperparameters on 0.01% of data manually in a supervised manner [15]. This is analogous to how humans learn quickly from few samples. Hyperparameters depend on the application task [3]. On the other hand, they help in controlling the hardness of a benchmark, which can be leveraged in an active learning setting to develop dynamic benchmarks.

Results: DQI component colors across settings are correctly predicted according to AFLite catego-

[4]See Supplemental Material: Interface Design for interface intuitions and description

[5] See Supplemental Material: Evaluation, for details across all components and analyses.

rization of good and bad splits on an average [6] of 10/12 times in SNLI, 5/8 times in SQUAD 2.0 and Story CLOZE, and 7/12 times in MNLI [5] as illustrated in Table 2. We convert SQUAD 2.0 and Story CLOZE into NLI format, with *answer* and *ending* corresponding to *hypothesis*, and *context* and *story* corresponding to *premise*, respectively.

Dataset	SNLI											
Sample ID	S7	S8	S9	S10	S11	S12	S5	S6	S3	S4	S1	S2
Split	Good						Bad					
Label	Entailment		Neutral		Contradiction		Entailment		Neutral		Contradiction	
DQI Color												

Dataset	Story CLOZE							
Sample ID	S1	S2	S3	S4	S5	S6	S7	S8
Split	Good				Bad			
Label	True		False		True		False	
DQI Color								

Dataset	MNLI											
Sample ID	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12
Split	Good						Bad					
Label	Entailment		Neutral		Contradiction		Entailment		Neutral		Contradiction	
DQI Color												

Dataset	SQUAD 2.0							
Sample ID	S1	S2	S3	S4	S5	S6	S7	S8
Split	Good				Bad			
Label	True		False		True		False	
DQI Color								

Table 2: Evaluating VAIDA over the most sensitive DQI component, Intra-Sample Word Similarity. Successes: green/orange for good, red/orange for bad split. Failures: red for good, green for bad split.

Analysis: False positives and false negatives can be attributed to the limitation of AFLite in incorrectly classifying samples [15]. Additionally, we have two observations: (i) VAIDA's prediction accuracy decreases as the artifact level in a dataset decreases. (ii) The values of most DQI sub-components do not change significantly (<25% of the time) after adding samples in both categories. However, it changes considerably (>60% of the time) across two sub-components: Intra-sample word overlap and word similarity, both of which belong to the fifth component of DQI. This can again be explained by AFLite's sensitivity towards word overlap [15].

5.2 Expert Review

We present an initial prototype of our tool, to a set of three researchers with expertise in NLP and knowledge of data visualization, in order to judge the interface design. For each expert, the crowdworker interface and then analyst interfaces were demoed. Participants could ask questions and make interaction/navigation decisions to facilitate a natural user experience. All the experts appreciated the easily interpretable traffic-signal color scheme and found the organization of the interfaces—providing separate detailed views within the analyst workflow– a way to prevent cognitive overload (too much information on one screen) while allowing multi-granular analysis; this would help in classifying samples of middling quality as benchmark size increases with relative ease.

5.3 User Study

Setup: We approach several software developers, testing managers, and undergraduate/ graduate students. Based on their domain familiarity (in NLP and visualization), we split them into 23 crowdworkers and 8 analysts for constructing NLI samples, given premises. There are 100 high quality samples in the system at the time each participant participates in each round. Their experience is evaluated using NASA Task Load Index[7][7], where each task is scored in a 100-points range, with 5-point steps. To conduct an ablation study, we introduce modules one at a time (and finally the complete system) to all user classes as follows: (i) Crowdworkers— conventional crowdsourcing, traffic signal feedback, AutoFix, all, and (ii) Analysts— conventional analysis, traffic signal feedback, visualizations, TextFooler, all. For both user categories, a preliminary walkthrough of panels using 2 fixed samples– chosen randomly from the set used for the case study with SNLI– is conducted for each round of the study (Figure 4(a)).

Analysis: Figure 4(b) summarizes study results, averaged over all user responses. The users are presented with system modules in the order listed, and are asked to report scores relative to the original score they assign the conventional crowdsourcing/analysis approaches; at the end of each round, they are also asked for their comments[8].

Crowdworkers: Traffic signal feedback initially increases time (29.2%) and effort (65%) required to create high quality samples, as users have to correct them. However they are more confident (performance– 23.1%) of sample quality. AutoFix usage causes an unexpected increase in effort (10%) and frustration (77.8%), as users do not fully trust recommendations without visual feedback.

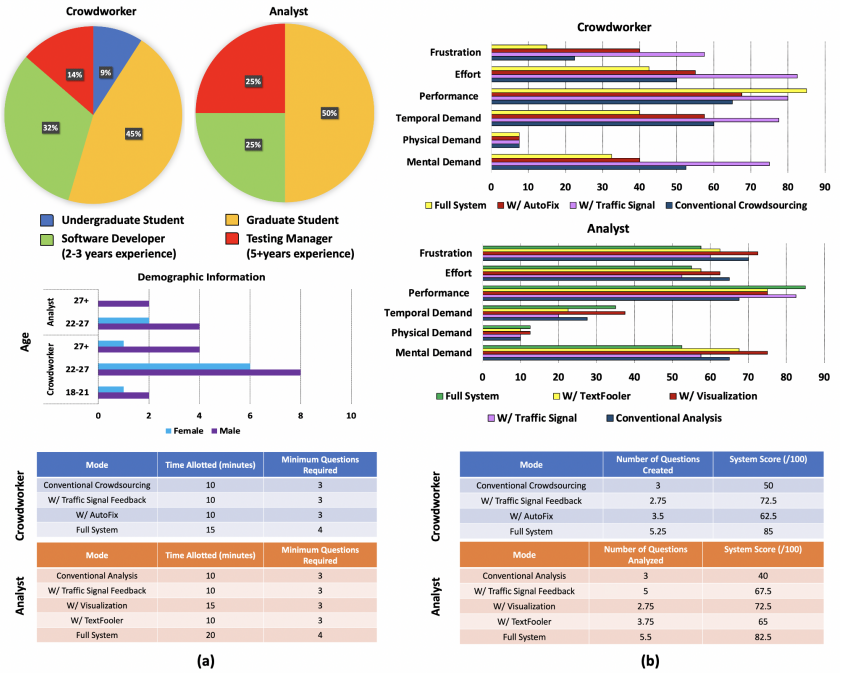
[6] We run (i) once and (ii) 10 times.

[7] See Supplemental Material: User Study for more details.

[8] We aggregate comment analysis here, see Supplemental Material: Expert and User Comments for quotes.

The drastic improvement over all aspects (frustration– 33.3%, mental demand– 38.1%, temporal demand– 33.3%, effort– 15%, average decrease in difficulty– 29.7%, performance– 30.8%) in the case of using the full system is in line with this observation. The number of questions created per round (traffic signal– 8.3%, AutoFix– 16.7%, full system– 75%) as well as system scores (traffic signal– 45%, AutoFix– 25%, full system– 70%) also follows this trend, across all types of crowdworkers.

Analysts: Analysts find the task easier (effort– 19.3%, performance– 22.2%) with traffic signal feedback, as quality is clearly marked.



Crowdworker	Mode	Time Allotted (minutes)	Minimum Questions Required
	Conventional Crowdsourcing	10	3
	W/ Traffic Signal Feedback	10	3
	W/ AutoFix	10	3
	Full System	15	4

Analyst	Mode	Time Allotted (minutes)	Minimum Questions Required
	Conventional Analysis	10	3
	W/ Traffic Signal Feedback	10	3
	W/ Visualization	15	3
	W/ TextFooler	10	3
	Full System	20	4

Crowdworker	Mode	Number of Questions Created	System Score (/100)
	Conventional Crowdsourcing	3	50
	W/ Traffic Signal Feedback	2.75	72.5
	W/ AutoFix	3.5	62.5
	Full System	5.25	85

Analyst	Mode	Number of Questions Analyzed	System Score (/100)
	Conventional Analysis	3	40
	W/ Traffic Signal Feedback	5	67.5
	W/ Visualization	2.75	72.5
	W/ TextFooler	3.75	65
	Full System	5.5	82.5

Figure 4: (a) User Study Setup, (b) User Study Results (averaged over all responses)

When analysts are shown the visualization interfaces, they are explicitly taught to differentiate the traffic signal colors in the visualizations as being indicative of how the sample affects the overall dataset quality, i.e., the colors in different component views represent individual terms of the components calculated over the whole dataset (analysts can toggle between the states of original dataset and new sample addition). We find that users initially find this more difficult to do (mental demand– 15.4%, temporal demand– 36.4%, frustration– 3.5%), though they agree that it improves their judgement of quality (performance– 11.1%). Analysts averaged behavior on TextFooler models the conventional approach quite closely, as analysts are seen to have a tendency to send all samples that are unclear to TextFooler immediately. With the full system, analysts also report improvement in all aspects (average decrease in difficulty– 12.1%), particularly mental demand (19.2%) and performance (26%), considering that the system increases the likelihood of a low hypothesis baseline. The visualization usage also improves, as analysts learn component relationships. Altogether, sample evaluation by analysts increases (full system– 83.3%), following this trend, and analysts are more assured of their performance (full system score– 106.25%).

Learning Curve: At the end of the study, all users are asked the following: *"What do you think high quality means?"* We find that users are able to distinguish certain patterns that promote higher quality, such as keeping sentence length appropriate and uniform across labels (not too long/short), using complex phrasing ('not bad')/gender information/modifiers across labels, and decreasing premise-hypothesis word overlap; they also do not display undesirable behavior like tweaking previously submitted samples just to create more.

User Education: We also conduct a variation of the study where a subset of participants (7 crowdworkers and 2 analysts) agreed to create/ analyze samples, for varying numbers of pre-accepted samples (Figure 5), in only the full system condition. In general, as the number of samples increases, the proportion of red or mixed samples also increases, and those green decreases. We find that when beginning from the cold start condition, as the sample number increases, due to their familiarity with the system, both crowdworkers and analysts are able to leverage the system better to avoid

red samples. However, when participants are directly started in situations with > 500 samples in the system, their unfamiliarity with the system initially causes a steepening of the learning curve compared to the cold start condition; this also tapers and saturates more slowly than cold start as the users gain experience. In the case of cold start, we find that users who create $\sim$50 samples report lesser reliance on AutoFix as they get better at creating higher quality samples; those who analyze $\sim$75 samples use TextFooler more efficiently as they understand how to deal with samples of middling quality better.

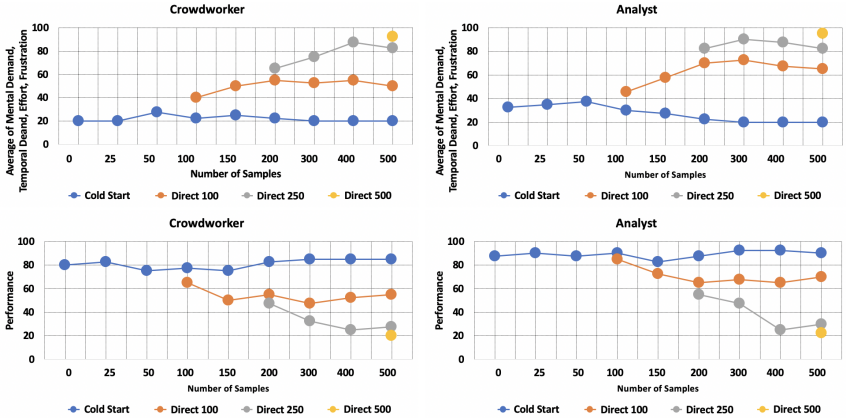


Figure 5: User education curves. Cold start has no pre-existing samples, and direct-n has n pre-existing samples. Mental Demand, Temporal Demand, Frustration, and Effort are averaged, Physical Demand is ignored. Performance is plotted separately as it shows differing behavior than the others.

6 Discussions and Future Work

VAIDA decreases mental demand, as well as the pressure (temporal demand, frustration) of creation and validation. Users report greater satisfaction with their work (performance); we also observe that users process more than the required samples with the full system in our user study due to the lower effort expended. This implies possible higher crowdworker retention and engagement. The use of AutoFix and TextFooler initially compensates for learning/decision fatigue; participants become less reliant on these over time and report less fatigue, indicating an understanding of data quality and its relevant features. We intend to integrate VAIDA with an actual crowdsourcing framework, and create a high quality benchmark, which we will evaluate using DQI and other metrics[6]. In our ablation study, we introduce modules in a fixed order to users, as per the patterns of usage preferred by the experts. We will compare this directly with the effectiveness of explicit user training [19] on patterns that decrease quality, as well as user training using an alternate workflow to see if/how user strategy changes. While both AutoFix and TextFooler can potentially result in the production of artificial sentences, as human intervention is allowed, we expect the final sample to be natural for a larger benchmark; we observe this in our user study. The most sensitive DQI component is found to involve word overlap; AFLite's removed samples also exhibit larger word overlap among other artifacts [6]. We will modify the n-gram, and sentence related DQI subcomponents to increase the range of bias captured.

7 Conclusion

We propose VAIDA, a paradigm to address benchmark bias, by integrating human-in-the-loop sensemaking with continuous feedback from a data quality metric, DQI. We design complementary workflows for both crowdworkers and analysts, to create new samples, evaluate them for the existence of artifacts, and review/repair samples to ensure the overall benchmark quality. We also develop AutoFix for automated data repair, and design a mechanism for adversarial tranformation to improve data quality by leveraging TextFooler. We also construct several visualization interfaces to analyze quality considerations at multiple granularities. VAIDA is evaluated with several case studies, a set of expert reviews– which provide qualitative feedback about the overall workflow experience– and a user study with NASA TLX. We find that usage of VAIDA decreases mental demand, temporal demand, effort, and frustration of crowdworkers (29.7%) and analysts(12.1%); it increases performance by 30.8% and 26% respectively, and educates users on data quality. VAIDA demonstrates a novel, dynamic approach for building benchmarks and mitigating bias, and serves as a starting point for the next generation of benchmarks in AI.

References

[1] Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*, 2015.

[2] Ronan Le Bras, Swabha Swayamdipta, Chandra Bhagavatula, Rowan Zellers, Matthew E Peters, Ashish Sabharwal, and Yejin Choi. Adversarial filters of dataset biases. *arXiv preprint arXiv:2002.04108*, 2020.

[3] Christopher Clark, Mark Yatskar, and Luke Zettlemoyer. Don't take the easy way out: Ensemble based methods for avoiding known dataset biases. *arXiv preprint arXiv:1909.03683*, 2019.

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[5] Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. Drop: A reading comprehension benchmark requiring discrete reasoning over paragraphs. *arXiv preprint arXiv:1903.00161*, 2019.

[6] Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel R Bowman, and Noah A Smith. Annotation artifacts in natural language inference data. *arXiv preprint arXiv:1803.02324*, 2018.

[7] Sandra G Hart. Nasa-task load index (nasa-tlx); 20 years later. In *Proceedings of the human factors and ergonomics society annual meeting*, volume 50, pages 904–908. Sage publications Sage CA: Los Angeles, CA, 2006.

[8] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*, 2016.

[9] Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. Is bert really robust? natural language attack on text classification and entailment. *arXiv preprint arXiv:1907.11932*, 2019.

[10] Divyansh Kaushik, Eduard Hovy, and Zachary C Lipton. Learning the difference that makes a difference with counterfactually-augmented data. *arXiv preprint arXiv:1909.12434*, 2019.

[11] Divyansh Kaushik and Zachary C Lipton. How much reading does reading comprehension require? a critical investigation of popular benchmarks. *arXiv preprint arXiv:1808.04926*, 2018.

[12] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240, 2020.

[13] Nitika Mathur, Tim Baldwin, and Trevor Cohn. Tangled up in bleu: Reevaluating the evaluation of automatic machine translation evaluation metrics. *arXiv preprint arXiv:2006.06264*, 2020.

[14] R Thomas McCoy, Ellie Pavlick, and Tal Linzen. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. *arXiv preprint arXiv:1902.01007*, 2019.

[15] Swaroop Mishra, Anjana Arunkumar, Bhavdeep Singh Sachdeva, Chris Bryan, and Chitta Baral. Dqi: A guide to benchmark evaluation. *arXiv: Computation and Language*, 2020.

[16] Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. Hypothesis only baselines in natural language inference. *arXiv preprint arXiv:1805.01042*, 2018.

[17] Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for squad. *arXiv preprint arXiv:1806.03822*, 2018.

[18] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*, 2016.

[19] Paul Roit, Ayal Klein, Daniela Stepanov, Jonathan Mamou, Julian Michael, Gabriel Stanovsky, Luke Zettlemoyer, and Ido Dagan. Crowdsourcing a high-quality gold standard for qa-srl. *arXiv preprint arXiv:1911.03243*, 2019.

[20] Rachel Rudinger, Chandler May, and Benjamin Van Durme. Social bias in elicited natural language inferences. In *Proceedings of the First ACL Workshop on Ethics in Natural Language Processing*, pages 74–79, 2017.

[21] Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *arXiv preprint arXiv:1907.10641*, 2019.

[22] Roy Schwartz, Maarten Sap, Ioannis Konstas, Li Zilles, Yejin Choi, and Noah A Smith. The effect of different writing tasks on linguistic style: A case study of the roc story cloze task. *arXiv preprint arXiv:1702.01841*, 2017.

[23] Antonio Torralba and Alexei A Efros. Unbiased look at dataset bias. In *CVPR 2011*, pages 1521–1528. IEEE, 2011.

[24] Adina Williams, Nikita Nangia, and Samuel R Bowman. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*, 2017.

[25] Sheng Zhang, Rachel Rudinger, Kevin Duh, and Benjamin Van Durme. Ordinal common-sense inference. *Transactions of the Association for Computational Linguistics*, 5:379–395, 2017.

[26] Yuan Zhang, Jason Baldridge, and Luheng He. Paws: Paraphrase adversaries from word scrambling. *arXiv preprint arXiv:1904.01130*, 2019.