

# Real-Time Visual Feedback to Guide Benchmark Creation: A Human-and-Metric-in-the-Loop Workflow

Anjana Arunkumar<sup>1</sup>  
aarunku5@asu.edu

Swaroop Mishra<sup>1</sup>  
srmishr1@asu.edu

Bhavdeep Sachdeva<sup>1</sup>  
bssachde@asu.edu

Chitta Baral<sup>1</sup>  
cbaral@asu.edu

Chris Bryan<sup>1</sup>  
cbryan16@asu.edu

<sup>1</sup>Arizona State University

## Abstract

Recent research has shown that language models exploit ‘artifacts’ in benchmarks to solve tasks, rather than truly learning them, leading to inflated model performance. In pursuit of creating *better benchmarks*, we propose VAIDA, a novel benchmark creation paradigm for NLP, that focuses on *guiding crowdworkers*, an under-explored facet of addressing benchmark idiosyncrasies. VAIDA facilitates sample correction by providing real-time visual feedback and recommendations to improve sample quality. Our approach 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 evaluate via expert review and a user study with NASA TLX. We find that VAIDA decreases effort, frustration, mental, and temporal demands of crowdworkers and analysts, simultaneously increasing the performance of both user groups with a **45.8%** decrease in the level of artifacts in created samples. As a by-product of our user study, we observe that created samples are adversarial across models, leading to decreases of **31.3%** (BERT), **22.5%** (RoBERTa), **14.98%** (GPT-3 fewshot) in performance.<sup>1</sup>

## 1 Introduction

Researchers invest significant effort to create benchmarks in machine learning, including ImageNet (Deng et al., 2009), SQUAD (Rajpurkar et al., 2016), and SNLI (Bowman et al., 2015), as well as to develop models that solve them. *Can we rely on these benchmarks?* A growing body of recent research (Schwartz et al., 2017; Poliak et al., 2018; Kaushik and Lipton, 2018) is revealing that models exploit spurious bias/artifacts—unintended correlations between input and output (Torralba and Efros, 2011) (e.g. the word ‘not’ is associated with

the label ‘contradiction’ in Natural Language Inference (NLI) (Gururangan et al., 2018))—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 (Bras et al., 2020; Mishra et al., 2020a; McCoy et al., 2019; Zhang et al., 2019; Larson et al., 2019b; Sakaguchi et al., 2019; Hendrycks and Gimpel, 2016). This begs the question: *Shouldn’t ML researchers consequently focus on creating ‘better’ datasets rather than developing increasingly complex models on bias-laden benchmarks?*

Deletion of samples based on bias baseline reports—hypothesis-only baseline in NLI (Dua et al., 2019)—and mitigation approaches such as AFLite (Sakaguchi et al., 2019) (adversarial filtering which deletes targeted data subsets), (Clark et al., 2019; Kaushik et al., 2019), have 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 adequate feedback to learn what constitutes high-quality data— and so have additional overhead due to the manual effort involved in sample creation/validation. Furthermore, (Parmar et al., 2022a) show that biased samples are created even when crowdworkers are provided with an initial set of annotation instructions. One potential solution to these problems is continuous, *in situ* feedback about artifacts while benchmark data is being created. *To our knowledge, there are no approaches that provide real-time 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 in real-time. VAIDA supports artifact identification

<sup>1</sup>A video description of VAIDA, generated samples, and detailed analyses are available in the Supplemental Material.

Component Name	DQI Implication	VAIDA Usage	Artifacts Evaluated
1. Vocabulary	Ambiguity and diversity of a dataset’s language	Does the sample contribute new words?	Sample Length (Wallace et al., 2019), New Words Introduced (Yaghouh-Zadeh-Fard et al., 2020; Larson et al., 2020), Jaccard Index between n-grams (Larson et al., 2019a)
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?	N-gram overlap (Wallace et al., 2019; Yaghouh-Zadeh-Fard et al., 2020), Mean-IDF (Stasaski et al., 2020)
3. Inter-Sample STS	Syntactic, semantic, and pragmatic sentence parsing	How similar is the hypothesis to all other premises or hypotheses?	Multi-hop reasoning (Wallace et al., 2019), Similarity and overlap (Yaghouh-Zadeh-Fard et al., 2020), Diversity (Larson et al., 2019a)
4. Intra-Sample Word Similarity	Word overlap and similarity within sample statements	How similar are all words within a sample?	Coreference Resolution, Multi-hop reasoning (Wallace et al., 2019), Word Overlap (Larson et al., 2020)
5. Intra-Sample STS	Phrase/sentence level overlap within a sample	How similar is the hypothesis to the premise?	N-gram repetition and overlap (Yaghouh-Zadeh-Fard et al., 2020)
6. N-gram Frequency per Label	Distribution of samples according to annotation	Is the hypothesis too obvious for the system?	Logic and Calculations (Wallace et al., 2019), Diversity (Larson et al., 2019a), Outliers, Entropy (Stasaski et al., 2020)
7. Inter-Split STS	Optimal similarity between train and test samples	Is the sample too similar to an existing sample?	Entity Distractors, Novel Clues (Wallace et al., 2019), Coverage (Larson et al., 2019a)

Table 1: Language properties considered in DQI that indicate artifact presence, their interpretation in VAIDA, and corresponding methods used in crowdsourcing pipeline evaluation; STS: semantic textual similarity.

and resolution, implicitly educating *crowdworkers* and *analysts* on data quality. (ii) We evaluate VAIDA empirically through expert review and a user study to understand the cognitive workload it imposes. The results indicate that VAIDA decreases mental demand, temporal demand, effort, and frustration of crowdworkers (31.1%) and analysts (14.3%); it increases their performance by 34.6% and 30.8% respectively, and educates crowdworkers on how to create *high-quality* samples. Overall, we see a 45.8% decrease in the presence of artifacts in created samples. (iii) Even though our main goal is to reduce artifacts in samples, we observe that samples created in our user study are adversarial across language models with performance decreases of 31.3% (BERT), 22.5% (RoBERTa), and 14.98% (GPT-3 fewshot).

## 2 Related Work

This work sits at the intersection of two primary areas: (1) *visual analysis of data quality* (higher presence of artifacts indicates lower quality), and (2) *development of a novel data collection pipeline*.<sup>2</sup>

### 2.1 Sample Quality and Artifacts

Data Shapley (Ghorbani and Zou, 2019) has been proposed as a metric to quantify the value of each training datum to the predictor performance. However, the metric might not signify bias content, as the value of training datum is quantified based on predictor performance, and biases might favor the predictor. Moreover, this approach is model and task-dependent. VAIDA uses DQI (Data Quality Index), proposed by (Mishra et al., 2020b), to: (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. Table 1 broadly defines DQI components, along with their interpretation in VAIDA, and juxtaposes them against evaluation methods used in prior works on crowdsourcing pipelines, as discussed in 2.2. (Wang et al., 2020) propose a tool for measuring and mitigating artifacts in image datasets.

### 2.2 Crowdsourcing Pipelines

Several pipelines have been proposed to handle various aspects of artifact presence in samples.

**Adversarial Sample Creation:** Pipelines such as Quizbowl (Wallace et al., 2019) and Dynabench (Kiela et al., 2021), highlight portions of text from input samples during crowdsourcing, based on how important they are for model prediction; this prompts users to alter their samples, and produce samples that can fool the model being used for evaluation (Talmor et al., 2022). While these provide more focused feedback compared to adversarial pipelines like ANLI (Nie et al., 2019), which do not provide explicit feedback on text features, adversarial sample creation is contingent on performance against a specific model (Quizbowl for instance is evaluated against IR and RNN models, and may therefore not see significant performance drops against more powerful models). Additionally, such sample creation might introduce new artifacts over time into the dataset and doesn’t always correlate with high quality— for instance, a new entity introduced to fool a model in an adversarial sample might be the result of insufficient inductive bias, though reducing the level of spurious bias.

A similar diagnostic approach is followed for

<sup>2</sup>Detailed related work is in the Supplemental Material.

unknown unknown identification– i.e., instances for which a model makes a high-confidence prediction that is incorrect. (Attenberg et al., 2015) and (Vandenhof, 2019) propose techniques to identify UUs, in order to discover specific areas of failure in model generalization through crowdsourcing. The detection of these instances is however, model-dependent; VAIDA addresses the occurrence of such instances by comparing sample characteristics between different labels to identify (and resolve) potential artifacts and/or under-represented features in created data.

**Promoting Sample Diversity:** Approaches focusing on improving sample diversity have been proposed, in order to promote model generalization. (Yaghouh-Zadeh-Fard et al., 2020) use a probabilistic model to generate word recommendations for crowdworker paraphrasing. (Larson et al., 2019a) propose retaining only the top  $k\%$  of paraphrase samples that are the greatest distance away from the mean sentence embedding representation of all collected data. These ‘outlier’ samples are then used to seed the next round of paraphrasing. (Larson et al., 2020) iteratively constrain crowdworker writing by using a taboo list of words, that prevents the repetition of over-represented words, which are also a source of spurious bias. Additionally, (Stasaski et al., 2020) assess the new sample’s contribution to the diversity of the entire sub-corpus.

**Controlled Dataset Creation:** Previous work (Roit et al., 2019) 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.

As shown in Table 1, DQI encompasses the aspects of artifacts studied by the aforementioned works; it further quantifies the presence of many more inter and intra-sample artifacts,<sup>3</sup> and provides a one-stop solution to address artifact impact on multiple fronts. VAIDA leverages DQI to identify artifacts, and further focuses on educating crowdworkers on exactly ‘why’ an artifact is undesirable,

<sup>3</sup>See Supplemental Material for details on artifacts that DQI identifies.

as well as the impact its presence will have on the overall corpus. It is also easily extensible to incorporate additional metrics such as quality control measures (Ustalov et al., 2021), enabling benchmarking evaluation in a reproducible manner. This is in contrast to the implicit feedback provided by word recommendation and/or highlighting in prior works, based on a static set of metrics– VAIDA facilitates the systematic elimination of artifacts without the unintentional creation of new artifacts, something that has hitherto remained unaddressed.

### 3 Modules

In this section, we describe VAIDA’s important backend processes.

**DQI:** DQI can be expressed as a quality metric that examines different sources of artifacts in text, by scoring samples along 7 different components. We use DQI in order to demonstrate VAIDA’s ability to cover multiple facets of artifact creation, although VAIDA is metric agnostic. VAIDA uses an intuitive traffic signal color coding (**high**>>**moderate**>>**low**) to indicate levels of artifacts (i.e., quality) in samples. Hyperparameters for color mapping with respect to DQI component values depend on (i) the application type, and (ii) characteristics of pre-existing samples present in the corpus at the time of new sample creation.<sup>4</sup> For instance, when recreating SNLI (Bowman et al., 2015) with VAIDA, we tune hyperparameters separating the boundary between red, yellow, and green flags on 0.01% of the SNLI training dataset manually in a supervised manner (Mishra et al., 2020b).

**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 1. Given a premise, hypothesis, and DQI values for the hypothesis, AutoFix sequentially masks each word in the hypothesis and ranks words based on their influence on model output, i.e., their importance. The hypothesis word of highest importance is replaced, to achieve at least **moderate** quality. DQI hence controls the amount and aspect of changes made by AutoFix. By incrementally changing the sample, users can understand how and why their sample is being modified and how DQI values are affected.

**TextFooler:** From an analyst’s perspective,

<sup>4</sup>See Supplemental Material: Evaluation, for details across all DQI components, hyperparameter tuning, and analyses.

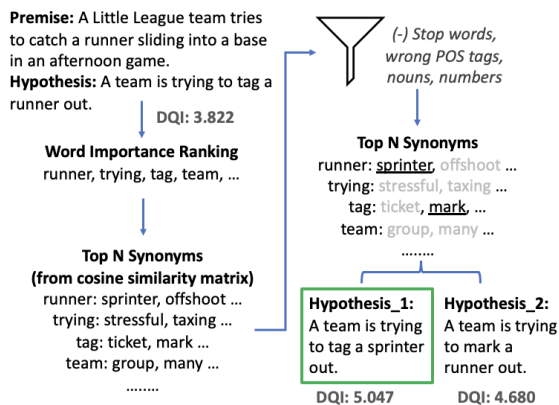


Figure 1: AutoFix Algorithm applied to an SNLI entailment sample, replacing one word per iteration. The DQI of the hypothesis changes from 3.822 to 5.047.

the quality of a submitted sample might be “too low” because (i) the crowdworker might not employ AutoFix appropriately, or (ii) there is a narrow acceptability range due to the criticality of the application domain, such as in BioNLP (Lee et al., 2020; Parmar et al., 2022b). We therefore implement TextFooler (Jin et al., 2019) to adversarially transform low-quality samples (instead of discarding them), to improve benchmark robustness, and ensure that crowdsourcing effort is not wasted. We initially use AFLite (Bras et al., 2020), 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 identify sample artifacts using DQI (see Figure 5).

#### 4 Interface Design and Workflow

VAIDA provides customized interfaces for both crowdworkers and analysts, as shown in Figures 2, 3 respectively.<sup>5</sup> We describe VAIDA’s workflow via a case study for sample creation (crowdworker) and review (analyst) in Figures 4,5.

The crowdworker interface provides instructions (A) to navigate through the panels and how to interpret feedback for created samples. Communication links for FAQs, and error reporting are also provided (F). Crowdworkers can then review the instructions for data creation (B)– here, we use the same instructions provided in the original SNLI crowdsourcing interface (b1).

**1. Sample Creation:** The premise field auto-populates with a caption from the Flickr30 corpus

<sup>5</sup>See Supplemental Material: Interface Design for interface intuitions and detailed descriptions, with full-resolution images.

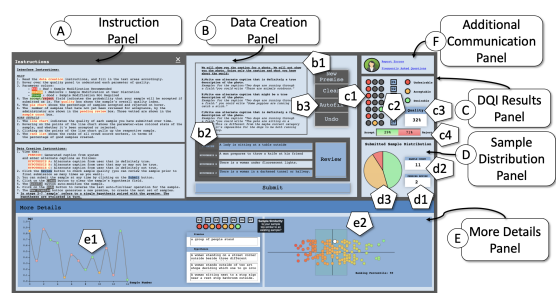


Figure 2: VAIDA’s crowdworker interface consists of six linked panels: (A) Instructions, (B) Data creation, (C) DQI results, (D) Sample distribution, (E) More details, and (F) Additional communication.<sup>5</sup>

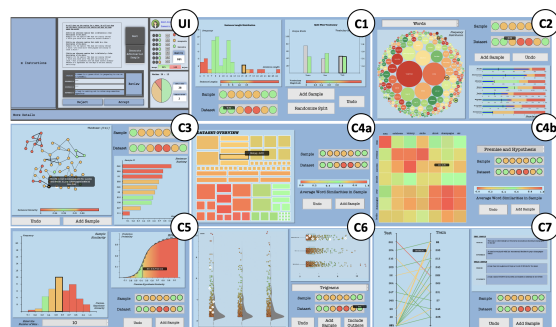


Figure 3: VAIDA provides a collection of interfaces for the analyst supporting detailed analysis and investigation of submitted samples and the overall benchmark.<sup>5</sup>

(b2). The crowdworker must create three hypotheses (for the entailment, neutral, contradiction labels) at a time, though they are reviewed individually.

**2. DQI Evaluation Feedback:** On clicking the review button, DQI feedback is shown in (C), for each component. Each colored circle (c1) indicates the level of artifacts present corresponding to each DQI component for the created sample; hovering displays a tooltip that explains and highlights (c2) the artifacts in the created sample, pertaining to the category of bias covered by that component. The overall sample quality (c3) is calculated, by averaging over the artifact percentiles of all 7 DQI components for the sample. By comparing sample quality with that of pre-existing samples, the probability that the sample will be accepted/rejected is shown (c4). The user can choose to either revise or submit the sample.

**3a. Sample Revision:** *Manual Revision* or *AutoFix* can be done. We illustrate the improvement of sample quality in this step in Figure 4.

**3b. Sample Submission:** After review (and potentially iterative DQI evaluations/sample revi-

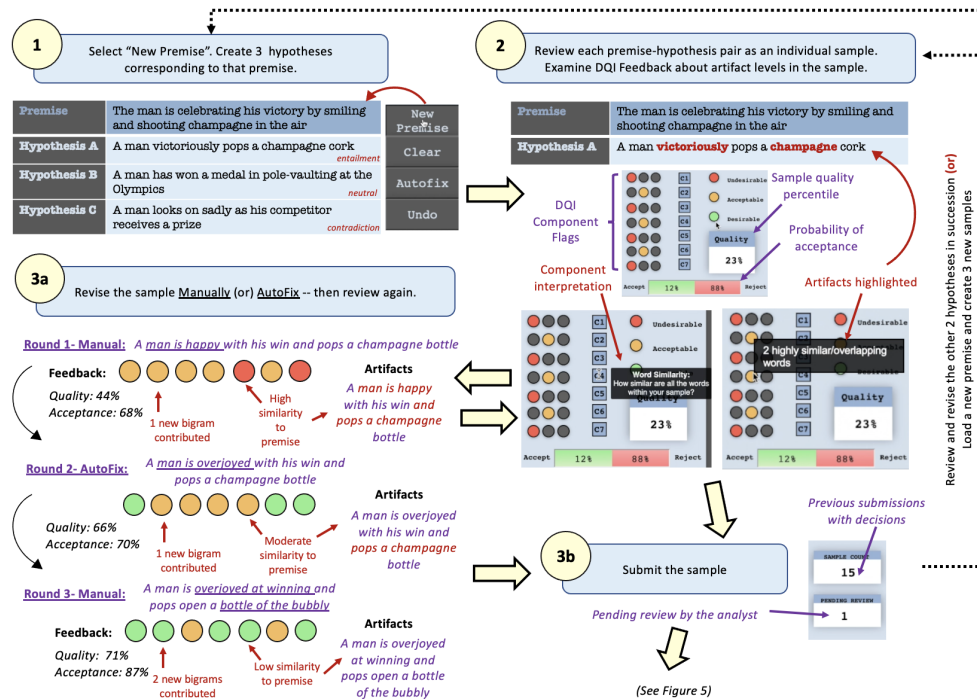


Figure 4: VAIDA workflow for the creation of a single sample by a crowdworker.

sions), the sample can be submitted for benchmark inclusion. On submission, a sample enters a *pending* state (**d1**) until review by the analyst.

While crowdworkers work within a single tightly-coordinated interface to create, submit, and review samples, analysts can navigate between a set of nine screens (Figure 3) to review samples in detail to make ‘accept’, ‘reject’, and ‘modification’ decisions, and to assess overall benchmark quality. Analysts review samples in batches of size 50.<sup>6</sup>

**4. Analyst Review:** Review can consist of several different operations by the analyst.

(i) *Direct Acceptance*– The home page (**UI**) for the analyst provides a view similar to the crowdworker interface, and allows the analyst to review the work of a single crowdworker. If the data quality is deemed to be sufficiently high by just viewing the DQI color flags and quality percentile, the analyst can directly accept the sample into the corpus from this screen, as shown in Figure 5.

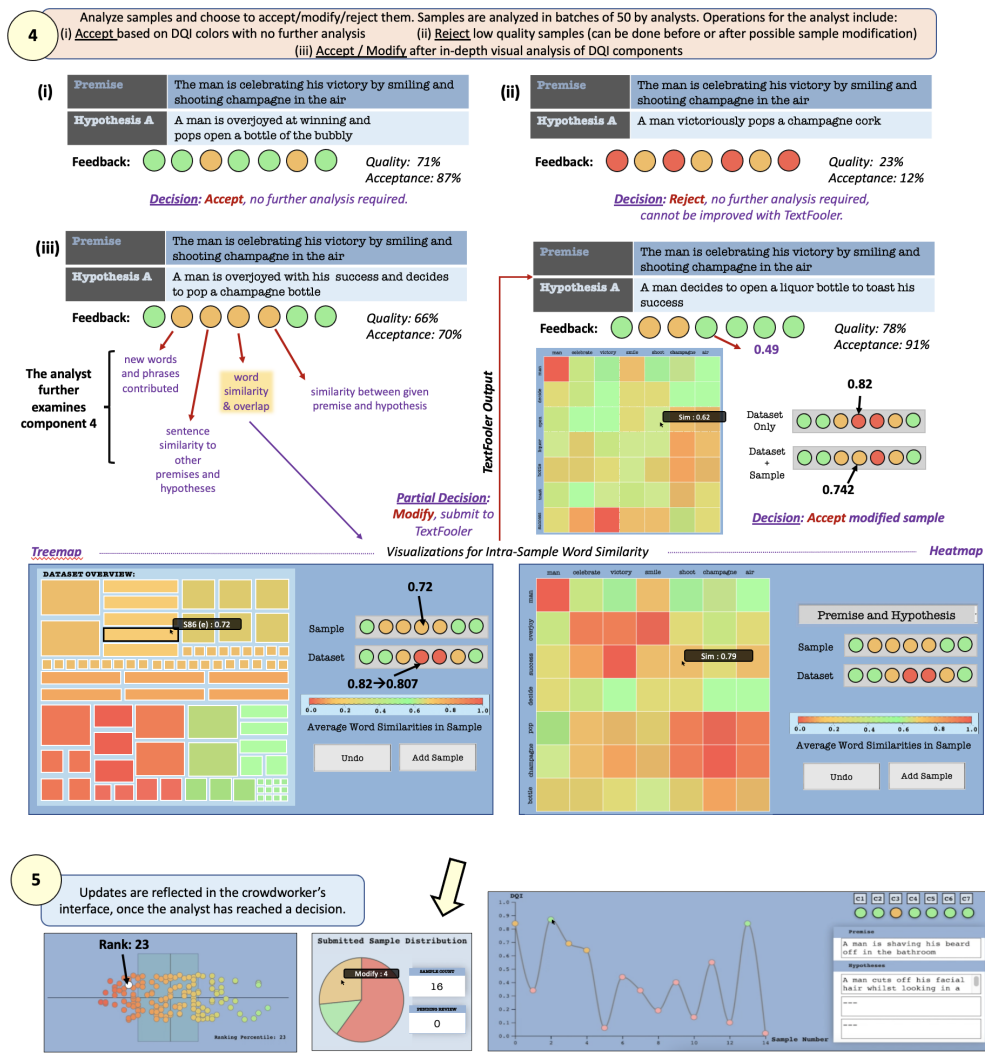
(ii) *Visual Analysis and Modification*– VAIDA provides several visualizations (**C1-C7**) to support detailed analysis and review of submitted samples, and to assess artifact presence (i.e., quality) in the

<sup>6</sup>DQI components 1, 2, 3, 6, and 7, gauge artifact presence relative to pre-existing samples. We observe given at least 50 pre-existing samples, DQI component values change by less than 5% for the overall dataset if another 50 samples are accepted.

overall benchmark. Each visualization allows the analyst to simulate how adding one or more submitted samples affects the benchmark’s quality.

For example, in Figure 5, we show the visualizations for analysis of DQI component 4 (C4), which deals with artifacts caused due to word similarity within a sample. For each sample in the corpus, the word similarities between all possible pairs of words are averaged; the samples are then hierarchically displayed as a treemap based on their DQI color mapping and average word similarity. We note that the color scale followed is bilinear, as while artifacts must be eliminated, there still needs to be a sufficient inductive bias for the sample to be solvable. The size of a rectangle indicates the distance of each sample from the average word similarity across all samples. The new sample is highlighted in the treemap with a black outline. In the example shown, we see that the dataset’s C4 value decreases slightly (0.807) when the new sample is added, though the flag color remains red.

Further examination of the new sample can be done, to establish why the sample has low effect on the dataset’s C4; the user clicks on the sample rectangle in the treemap, and is taken to a heat map view. The heatmap shows the similarity values between every word pair (from the premise/hypothesis/both) in a tooltip on hover, with the values mapped to the same color scale used for



**Rank: 23**

**Submitted Sample Distribution**

Figure 5: VAIDA workflow for the evaluation of a single sample by an analyst, and subsequent analyst feedback provided to the crowdworker.

the treemap . For instance, in Figure 5, the words ‘man’ and ‘champagne are repeated verbatim, and ‘overjoyed’, ‘smiling’, ‘celebrating’, ‘pop’, and ‘shoot’ also have similarity of [0.6–1]. This indicates that several words in the sample are too closely related and constitute artifacts.

Each such visualization is therefore individually tailored to represent a specific DQI component of interest, based on the linguistic features examined for artifact creation in that component. We further elaborate on the design intuitions and analysis conducted with all the component visualizations available to the analyst in the supplementary.

Post analysis, if the analyst feels that the submitted sample requires only a minor change (for instance, reshuffling or the addition of a single word) to warrant acceptance, then they can invoke the TextFooler module to transform the sample adver-

sarily, and then accept, thereby ensuring minimal data loss. In the case shown, TextFooler improves the sample’s C4 with most of the heatmap varying from 0.3-0.7. Due to this, the analyst decides to accept the updated sample.

**5. Analyst Feedback:** Once the analyst has reached a decision, the crowdworker sees updates (Figure 5) in (D) to the reviewed sample count (d2) – increases to 16– and the pie chart that indicates the distribution of actions taken by the analyst over all samples submitted by the crowdworker (d3). Additionally, in (E), the line chart (e1), which contains the history of previously submitted samples is updated. The x-axis denotes the sample number and the y-axis denotes the quality percentile (c3), of the corresponding sample. On click, the corresponding sample is loaded, along with its DQI component values and feedback to the crowdworker. The

crowdworker can also use the box plot (e2), to view their current rank, and choose to view the sample history of another crowd worker.

## 5 Evaluation

We evaluate VAIDA’s efficacy at providing real-time feedback to educate crowdworkers during benchmark creation using expert review and a user study. We also evaluate model performance (BERT (Devlin et al., 2018), RoBERTa (Liu et al., 2019), GPT-3 (fewshot) (Brown et al., 2020)) on data created with VAIDA during the user study.

### 5.1 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. For each expert, the two interfaces were demoed in a Pair-Analytics session (Arias-Hernandez et al., 2011). 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 further suggested that alternates be provided to account for color blindness) 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). A caveat of this would be the inability for an analyst to simultaneously juxtapose different component visualizations. It was also hypothesized that a learning curve of  $\sim 50$ – $60$  samples would be required for cohesive use of all system modules by both types of users; however, this would be offset by the eventual capability of users to deal with samples of middling quality based on their multi-granular feedback about artifact presence.

### 5.2 User Study

**Setup:** We approach several software developers, testing managers, and undergraduate/graduate students. Based on their domain familiarity (in NLP and visualization, rated from 1:novice-5:expert), 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 ablation round (Table 2). For both types of users, a preliminary walkthrough of the system configuration, using 2 fixed samples, is conducted for each round of the study (Figure 6). At the end of each

round, they are also asked for their comments.<sup>7</sup>

Configuration	Description	User
Conventional Crowdsourcing	No feedback or auto modification tools	C
Conventional Analysis	Manual review without feedback or modification tools	A
Traffic Signal Feedback	Color mapping based on DQI values	C, A
AutoFix	Incremental sample auto-modification functionality	C
TextFooler	Adversarial sample transformation functionality	A
Visualization	Data visualizations for in-depth exploration (also includes traffic signal feedback)	A
Full System	All modules and system functionalities	C,A

Table 2: System configurations used for randomly ordered ablation rounds presented to users functioning as crowdworkers (C) and analysts (A) in the study.

User experience (mental workload) is subjectively evaluated using NASA Task Load Index (Hart, 2006)<sup>7</sup> (NASA TLX); each dimension is scored in a 100-points range, with 5-point steps. Users are also asked to report overall ratings for each system configuration at the end of the study.

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	3
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	3	

Figure 6: User Study Setup— describes the timeframe and requirements of the user study over ablation rounds.

**Analysis:** Figure 7 summarizes study results,<sup>8</sup> averaged over all user responses, for different system configurations. The general trend across both crowdworkers and analysts is that there is: (i) significant improvement across all NASA TLX dimensions, (ii) increase in number of samples created/reviewed, and (iii) user ratings for the system, when comparing VAIDA to conventional interfaces. In the case of partial module availability, we find that the effectiveness of traffic signal feedback and visualizations is comparable. The use of AutoFix and TextFooler<sup>7</sup> is more prevalent initially, on creation/evaluation of a low or middling quality sample for users as: (i) crowdworkers find constructive

<sup>7</sup> See Supplemental Material: User Study for more details. We do aggregated analysis of comments, full quotes of comments are present in the Supplemental Material. We also have IRB approval to conduct this user study.

<sup>8</sup>All results are found to have  $p \leq 0.02$

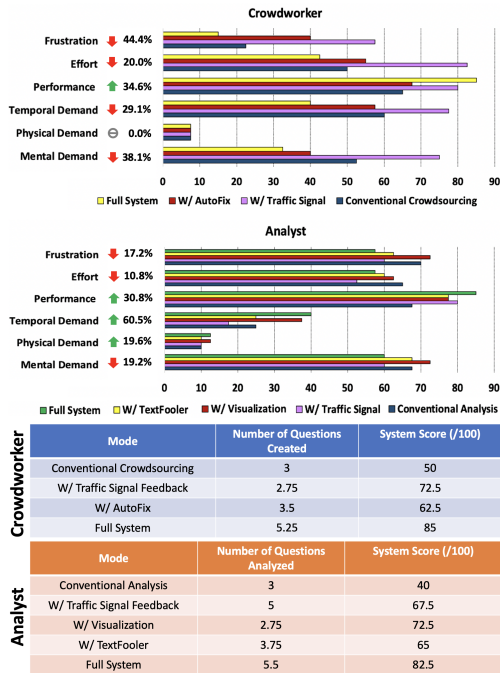


Figure 7: User Study Results– NASA TLX dimension scores, and user scoring of the configurations, averaged across participant responses, for each ablation round. Percentages shown indicate comparison of full system scores with respect to the conventional system.

sample modification more difficult initially, and (ii) analysts are initially unsure of how to deal with middling quality samples.

**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, decreasing premise-hypothesis word overlap; they also do not display undesirable behavior like tweaking previously submitted samples just to create more. We also find an overall decrease of **-45.8%** in the level of artifacts of created samples, across all rounds of ablation.

**User Education:** We also conduct a secondary study where a subset of participants (7 crowdworkers and 2 analysts) agreed to create/ analyze samples, for varying numbers of pre-accepted samples (Figure 8), in only the full system condition. We find that 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. This is attributed to: (a) an increased likelihood of samples of low/middling quality (more artifacts) being created (evinced by performance), and (b) lower impact of an individual sample on overall dataset quality. We also 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 in the cold start condition. These numbers increase by  $\sim 25\%$  when users start with 500 pre-existing samples.

### 5.3 Model Performance Results

We evaluate BERT and RoBERTa (trained on the full SNLI dataset), and GPT-3 (in fewshot setting) against the data created during the ablation rounds of the user study.<sup>9</sup> There are 100 high-quality samples ( $DQI > 0.7$ ) from the original SNLI dataset present in the system for the study<sup>10</sup>. This remains constant across all ablation rounds and for all users. 69 samples are created per ablation round during our study, across all users, for a total of 345 ( $69 \times 5$ ) samples that we evaluate with the models. Figure 9 shows the results for samples over each round of ablation. <sup>(10)</sup> In the case of TextFooler, samples are created using the ‘full system’ condition and then further modified using TextFooler by the analyst. The other sample sets are not modified by the analyst, and are directly accepted after evaluation.

We find that across all models, performance is lower when explicit quality feedback (via the traffic signal scheme) is provided, compared to the regular crowdsourcing condition. The highest drop is seen for BERT (**-20.66%**), while GPT-3 shows a lesser magnitude of performance decrease (**-12.89%**). Performance further decreases for all models when AutoFix is implemented, indicating the effectiveness of this module in seeding suggestions for sample improvement; the magnitude of performance loss follows BERT  $>>$  RoBERTa  $>>$  GPT-3. We can attribute this apparent variation in model robustness i.e., BERT  $<<$  RoBERTa  $<<$  GPT-3 as proportionate to increase in size of (i) the respective language models, and (ii) the pre-training corpora.

A significant decrease is seen in the full system and TextFooler conditions. Particularly, in the TextFooler round, performance sharply decreases

<sup>9</sup>We also create samples for the Story CLOZE dataset in the full system condition; see Supplementary Material

<sup>10</sup>The dataset is included in the Supplemental Material.



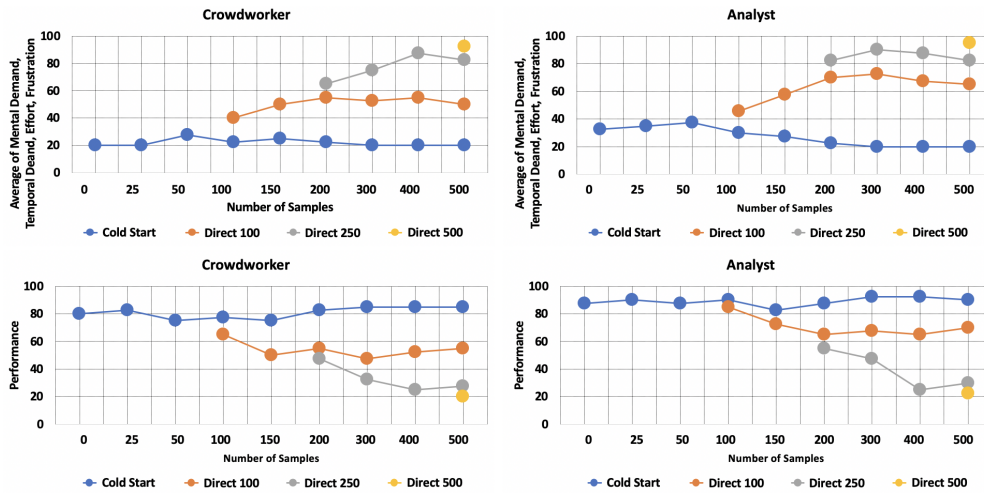


Figure 8: User education curves. Cold start has no pre-existing samples, and direct- $n$  has  $n$  pre-existing samples. Mental Demand, Temporal Demand, Frustration, and Effort are averaged, Physical Demand is ignored. Performance is plotted separately as it shows differing behavior from the others.

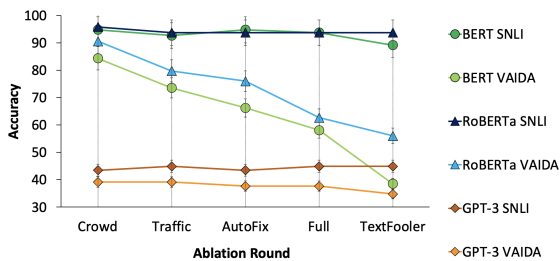


Figure 9: Model performance results for samples created during each ablation round of the user study.

for all models (-31.3% BERT, -22.5% RoBERTa, -14.98% GPT-3). Furthermore, in the TextFooler round, there is a -71.70% decrease in the level of sample artifacts. This indicates that crowdworkers and analysts are able to utilize VAIDA’s affordances to create more robust text samples.

## 6 Discussion and Conclusion

We propose VAIDA, a paradigm to address benchmark artifacts, by integrating human-in-the-loop sensemaking with continuous visual feedback. VAIDA uses several visualization interfaces to analyze quality considerations (based on artifact levels) at multiple granularities. While we do not explicitly address computational quality control or fairness consideration (though some aspects can be targeted by currently integrated metrics), since VAIDA is extensible to the incorporation of customized backend-metrics, we believe our paradigm can support multi-faceted benchmark evaluation.

In our usability evaluation, we see that users report greater satisfaction, and lower difficulty with

their work and system experience; this implies possible higher crowdworker retention and engagement. Additionally, in our study, we see that users effectively identify and avoid artifact patterns during sample creation. Based on our study results, we believe that a minimum of 30 annotators would be needed for large-scale data creation to ensure timely feedback (i.e., sample decision provided within 24 hours) to crowdworkers. Based on our secondary study, we believe analysts will exhibit increased performance and maintain satisfaction ratings in full-scale creation, as they will become well-versed in the nuances of bias for the data-creation task they are evaluating, as well as the visualizations being used. This, however, is contingent on restricting the visualization views to display only the 200–300 samples with closest artifact levels to the sample being evaluated.

Overall, samples created with VAIDA are found to not only of higher quality than achieved with conventional crowdsourcing, but are also seen to be adversarial across transformer models. This is also maintained across multiple task types— we additionally create StoryCLOZE (Schwartz et al., 2017) samples with VAIDA<sup>9</sup>. This was done independent of the study described in the paper, with 4 crowdworkers creating samples. However, for this, several interface features had to be changed, so we focus on reporting only NLI results in the main paper. VAIDA hence demonstrates a novel, dynamic approach for building benchmarks and mitigating artifacts, and serves as a starting point for the next generation of benchmarks in machine learning.

## Limitations

In future work, we intend to integrate VAIDA with an actual crowdsourcing framework, and run a full-scale data creation study to create a high-quality benchmark. Expanding to such a setup will require additional back-end engineering, to ensure that (i) timely and accurate feedback continues to be provided in real-time to crowdworkers, (ii) analysts are available on hand to process samples in a timely manner. This is out of scope for the current paper (e.g., it would require a significant budget), but we see our current work as a stepping stone in this direction. Additionally, studying the problem and designing the visualizations and real-time feedback mechanisms are essential steps before moving to large-scale evaluation; the novel affordances and designs are a necessary first step, and we believe they will be impactful to the NLP community.

Crowdworker retention and engagement in this full-scale setting also need to be evaluated, in order to better contextualize the learning curve associated with system usage and handling artifact creation, given an increasingly higher number of pre-existing system samples. Comparing this setup directly with the effect of in-depth user training (Roit et al., 2019) on artifact creation and review, prior to crowdsourcing, would also further help analyze and quantify if/how user strategy and performance changes during VAIDA usage.

Design modifications when creating different types of datasets will mainly require the redesigning of sample input fields, corresponding to the application and the type of metric used for artifact evaluation. However, in full-scale dataset creation, the visualization views for the analyst corresponding to different artifact types will have to be restricted to the  $\sim 300$  samples closest artifact levels, to the given sample being created, in order to facilitate scalable processing for analysts. Additionally, since visualization familiarity is required for the analyst to effectively review samples, the analysts may choose to streamline analysis by only using a subset of the provided visualization types in their version of the system, corresponding to the application domain.

## References

Richard Arias-Hernandez, Linda T. Kaastra, Tera M. Green, and Brian Fisher. 2011. [Pair analytics: Capturing reasoning processes in collaborative visual](#)

[analytics](#). In *2011 44th Hawaii International Conference on System Sciences*, pages 1–10.

Joshua Attenberg, Panos Ipeirotis, and Foster Provost. 2015. Beat the machine: Challenging humans to find a predictive model’s “unknown unknowns”. *Journal of Data and Information Quality (JDIQ)*, 6(1):1–17.

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

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

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

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

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

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

Amirata Ghorbani and James Zou. 2019. Data shapley: Equitable valuation of data for machine learning. *arXiv preprint arXiv:1904.02868*.

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

Sandra G Hart. 2006. 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.

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

- Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2019. Is bert really robust? natural language attack on text classification and entailment. *arXiv preprint arXiv:1907.11932*.
- Divyansh Kaushik, Eduard Hovy, and Zachary C Lipton. 2019. Learning the difference that makes a difference with counterfactually-augmented data. *arXiv preprint arXiv:1909.12434*.
- Divyansh Kaushik and Zachary C Lipton. 2018. How much reading does reading comprehension require? a critical investigation of popular benchmarks. *arXiv preprint arXiv:1808.04926*.
- Douwe Kiela, Max Bartolo, Yixin Nie, Divyansh Kaushik, Atticus Geiger, Zhengxuan Wu, Bertie Vidgen, Grusha Prasad, Amanpreet Singh, Pratik Ringshia, et al. 2021. Dynabench: Rethinking benchmarking in nlp. *arXiv preprint arXiv:2104.14337*.
- Stefan Larson, Anish Mahendran, Andrew Lee, Jonathan K Kummerfeld, Parker Hill, Michael A Laurenzano, Johann Hauswald, Lingjia Tang, and Jason Mars. 2019a. Outlier detection for improved data quality and diversity in dialog systems. *arXiv preprint arXiv:1904.03122*.
- Stefan Larson, Anish Mahendran, Joseph J Peper, Christopher Clarke, Andrew Lee, Parker Hill, Jonathan K Kummerfeld, Kevin Leach, Michael A Laurenzano, Lingjia Tang, et al. 2019b. An evaluation dataset for intent classification and out-of-scope prediction. *arXiv preprint arXiv:1909.02027*.
- Stefan Larson, Anthony Zheng, Anish Mahendran, Rishi Tekriwal, Adrian Cheung, Eric Guldán, Kevin Leach, and Jonathan K Kummerfeld. 2020. Iterative feature mining for constraint-based data collection to increase data diversity and model robustness. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8097–8106.
- Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.
- Nitika Mathur, Tim Baldwin, and Trevor Cohn. 2020. Tangled up in bleu: Reevaluating the evaluation of automatic machine translation evaluation metrics. *arXiv preprint arXiv:2006.06264*.
- R Thomas McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. *arXiv preprint arXiv:1902.01007*.
- Swaroop Mishra, Anjana Arunkumar, Chris Bryan, and Chitta Baral. 2020a. Our evaluation metric needs an update to encourage generalization. *ArXiv, abs/2007.06898*.
- Swaroop Mishra, Anjana Arunkumar, Chris Bryan, and Chitta Baral. 2022. A survey of parameters associated with the quality of benchmarks in nlp. *arXiv preprint arXiv:2210.07566*.
- Swaroop Mishra, Anjana Arunkumar, Bhavdeep Singh Sachdeva, Chris Bryan, and Chitta Baral. 2020b. Dqi: A guide to benchmark evaluation. *arXiv: Computation and Language*.
- Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2019. Adversarial nli: A new benchmark for natural language understanding. *arXiv preprint arXiv:1910.14599*.
- Mihir Parmar, Swaroop Mishra, Mor Geva, and Chitta Baral. 2022a. Don’t blame the annotator: Bias already starts in the annotation instructions. *arXiv preprint arXiv:2205.00415*.
- Mihir Parmar, Swaroop Mishra, Mirali Purohit, Man Luo, Murad Mohammad, and Chitta Baral. 2022b. In-boxbart: Get instructions into biomedical multi-task learning. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 112–128.
- Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. 2018. Hypothesis only baselines in natural language inference. *arXiv preprint arXiv:1805.01042*.
- Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.
- Paul Roit, Ayal Klein, Daniela Stepanov, Jonathan Mamou, Julian Michael, Gabriel Stanovsky, Luke Zettlemoyer, and Ido Dagan. 2019. Crowdsourcing a high-quality gold standard for qa-srl. *arXiv preprint arXiv:1911.03243*.
- Rachel Rudinger, Chandler May, and Benjamin Van Durme. 2017. Social bias in elicited natural language inferences. In *Proceedings of the First ACL Workshop on Ethics in Natural Language Processing*, pages 74–79.
- Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Winogrande: An adversarial winograd schema challenge at scale. *arXiv preprint arXiv:1907.10641*.
- David Schuff, Karen Corral, and Ozgur Turetken. 2011. Comparing the understandability of alternative data warehouse schemas: An empirical study. *Decision support systems*, 52(1):9–20.

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

Katherine Stasaski, Grace Hui Yang, and Marti A Hearst. 2020. More diverse dialogue datasets via diversity-informed data collection. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4958–4968.

Alon Talmor, Ori Yoran, Ronan Le Bras, Chandra Bhagavatula, Yoav Goldberg, Yejin Choi, and Jonathan Berant. 2022. Commonsenseqa 2.0: Exposing the limits of ai through gamification. *arXiv preprint arXiv:2201.05320*.

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

Dmitry Ustalov, Nikita Pavlichenko, Vladimir Losev, Iulian Giliuzev, and Evgeny Tulin. 2021. A general-purpose crowdsourcing computational quality control toolkit for python. *arXiv preprint arXiv:2109.08584*.

Colin Vanden Hof. 2019. A hybrid approach to identifying unknown unknowns of predictive models. In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, volume 7, pages 180–187.

Eric Wallace, Pedro Rodriguez, Shi Feng, Ikuya Yamada, and Jordan Boyd-Graber. 2019. Trick me if you can: Human-in-the-loop generation of adversarial examples for question answering. *Transactions of the Association for Computational Linguistics*, 7:387–401.

Angelina Wang, Arvind Narayanan, and Olga Russakovsky. 2020. Vibe: A tool for measuring and mitigating bias in image datasets. *arXiv preprint arXiv:2004.07999*.

Mohammad-Ali Yaghoub-Zadeh-Fard, Boualem Benatallah, Fabio Casati, Moshe Chai Barukh, and Shayan Zamanirad. 2020. Dynamic word recommendation to obtain diverse crowdsourced paraphrases of user utterances. In *Proceedings of the 25th International Conference on Intelligent User Interfaces*, pages 55–66.

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

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

## A Supplementary Material

The following information is included in the appendix.

- [Infrastructure Used](#)
- [Run-time Estimations](#)
- [Hyper Parameter](#)
- [Related Work](#)
- [DQI Components](#)
- [Interface Design Intuitions](#)
- [AutoFix and TextFooler Examples](#)
- [User Study](#)
- [Expert and User Comments](#)

Please refer to <https://github.com/aarunku5/VAIDA-EACL-2023.git> for:

- Video demos of VAIDA workflow
- Sample dataset generated during the ablation rounds of the user study
- DQI and Model Performance Results for User Study Samples
- DQI Evaluation: Artifact Case Study

### A.1 Infrastructure Used

In Section 3, we describe VAIDA’s flow by high level workflow and back-end processes(DQI, AutoFix, and TextFooler). Further, as discussed in Subsection 3.2 DQI can be used for quantifying artifact presence for the: i) overall benchmark, and ii) impact of new samples. Depending on the task at hand we run our experiments in different hardware settings. The DQI calculations run mostly using CPU, for new samples as well as overall samples. The AutoFix procedure, as explained in Subsection 3.2, gives the user assistance in improving quality on a per submission basis. Therefore that does not require high GPU intensive systems; we have provisions to shift execution to a GPU as well if necessary to speed up the process. For TextFooler the fine tuning of the model is run on "TeslaV100-SXM2-16GB"; CPU cores per node 20; CPU memory per node: 95,142 MB; CPU memory per core: 4,757 MB– this is not a necessity as code has been tested on lower configuration GPUs as well but we

have run our experiments in this setting. The attack part of the TextFooler requires more memory and we run that code on "Tesla V100-SXM2-32GB" compute Capability: 7.0 core Clock: 1.53GHz, coreCount: 80, device Memory Size: 31.75GiB device Memory Bandwidth: 836.37GiB/s.

## A.2 Run-time Estimations

The DQI calculations run on CPU (for real life setting purposes); for the approximate estimate of the time taken, we run experiments for fixed data size of 10K samples. If the DQI calculations are done to calculate the impact of individual new samples it take a couple of seconds. On the other hand, If we take the whole 10k size dataset it takes around 48 hours to complete the process on CPU. This whole process can be run in parallel to reduce the time taken to 16 hours.

The TextFooler part consists of two steps– the fine tuning part and attack part– for generating adversaries. For fine tuning models we use "TeslaV100-SXM2-16GB", and it takes 20-30 minutes to complete the process. For the attack part we use "Tesla V100-SXM2-32GB", which takes 2-3 hrs for completing 20k data samples. This estimate requires the cosine similarity matrix for word embeddings to be calculated before hand which takes around 1-2 hrs, but this step has to be done only if the word embeddings are modified. This is a rare task so we have kept this separated.

## A.3 Hyper Parameters

To look at the estimations of DQI and its variations, we have kept basic hyper-parameters fixed in the experiments. We keep the learning rate to  $1e-5$ , the number of epochs during the experiments are varied from 2-3, the per gpu train batch and eval batch sizes vary from 8-64 samples (the results shown are with respect to a batch size of 8), adam epsilon is set to  $1e-8$ , weight decay is set to 0, maximum gradient normalisation is set to 1, and maximum sequence length is set to 128. For TextFooler the the semantic similarity is fixed to 0.5 uniformly for all the experiments shown in this paper.

Additionally, the variations and range in the DQI parameters are dataset specific, i.e., hyperparameters depend on the application task. (Mishra et al., 2020b) design DQI as a generic metric to evaluate diverse benchmarks. However, the definitions of what constitutes high and low quality will vary depending on the application. For example, BiomedicalNLP might have lower tolerance levels for spu-

rious bias than General NLP. Another case is in water quality– cited as an inspiration for DQI by (Mishra et al., 2020b)– where the quality of water needed for irrigation is different than that of drinking or medicine. We can therefore say that the hyper-parameters in the form of boundaries separating high and low quality data (i.e., inductive and spurious bias) are dependent on applications.

## A.4 Related Work

### A.4.1 Sample Quality and Artifacts

Data Shapley (Ghorbani and Zou, 2019) has been proposed as a metric to quantify the value of each training datum to the predictor performance. However, the metric might not signify bias content, as the value of training datum is quantified based on predictor performance, and biases might favor the predictor. Moreover, this approach is model and task-dependent. VAIDA uses DQI (Data Quality Index), proposed by (Mishra et al., 2020b), to: (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. (Wang et al., 2020) concurrently propose a tool for measuring and mitigating artifacts in image datasets.

Data Shapley (Ghorbani and Zou, 2019) has been proposed as a metric to quantify the value of each training datum to the predictor performance. However, this approach is model and task dependent. More importantly, the metric might not signify bias content, as the value of training datum is quantified based on predictor performance, and biases might favor the predictor. VAIDA uses DQI (data quality index), proposed by (Mishra et al., 2020b), to: (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. The quality of individual features (aspects) of samples are evaluated based on decreasing presence of artifacts and increasing generalization capability. In a concurrent work (Wang et al., 2020), a tool for measuring and mitigating bias in image datasets has also been proposed. DQI estimates artifact presence by calculating seven component values corresponding to a set of language properties; along with their interpretation in VAIDA.

### A.4.2 Crowdsourcing Pipelines

**Adversarial Sample Creation:** Pipelines such as Quizbowl(Wallace et al., 2019) and Dynabench(Kiela et al., 2021), highlight portions of text from input samples during crowdsourcing,

based on how important they are for model prediction; this prompts users to alter their samples, and produce samples that can fool the model being used for evaluation. While these provide more focused feedback compared to adversarial pipelines like ANLI (Nie et al., 2019), which do not provide explicit feedback on text features, adversarial sample creation is contingent on performance against a specific model (Quizbowl for instance is evaluated against IR and RNN models, and may therefore not see significant performance drops against more powerful models). Additionally, such sample creation might introduce new artifacts over time into the dataset and doesn't always correlate with high quality— for instance, a new entity introduced to fool a model in an adversarial sample might be the result of insufficient inductive bias, though reducing the level of spurious bias.

A similar diagnostic approach is followed for unknown unknown identification— i.e., instances for which a model makes a high confidence prediction that is incorrect. (Attenberg et al., 2015) and (Vandenhof, 2019) propose techniques to identify UUs, in order to discover specific areas of failure in model generalization through crowdsourcing. The detection of these instances is however, model-dependent; VAIDA addresses the occurrence of such instances by comparing sample characteristics between different labels to identify (and resolve) potential artifacts and/or under-represented features in created data.

**Promoting Sample Diversity:** Approaches focusing on improving sample diversity have been proposed, in order to promote model generalization. (Yaghoub-Zadeh-Fard et al., 2020) use a probabilistic model to generate word recommendations for crowdworker paraphrasing. (Larson et al., 2019a) propose retaining only the top k% of paraphrase samples that are the greatest distance away from the mean sentence embedding representation of all collected data. These ‘outlier’ samples are then used to seed the next round of paraphrasing. (Larson et al., 2020) iteratively constrain crowdworker writing by using a taboo list of words, that prevents the repetition of over-represented words, which are also a source of spurious bias. Additionally, (Stasaski et al., 2020) assess the new sample’s contribution to the diversity of the entire sub-corpus.

DQI encompasses the aspects of artifacts studied by the aforementioned works; it further quantifies the presence of many more inter and intra-sample

artifacts, and provides a one stop solution to address artifact impact on multiple fronts. VAIDA leverages DQI to identify artifacts, and further focuses on educating crowdworkers on exactly ‘why’ an artifact is undesirable, as well as the impact its presence will have on the overall corpus. This is in contrast to the implicit feedback provided by word recommendation and/or highlighting in prior works— VAIDA facilitates the elimination of artifacts without the unintentional creation of new artifacts, something that has hitherto remained un-addressed.

### A.4.3 Task Selection and Controlled Dataset Creation

In this work, we demonstrate 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 probabilistic mutual information (PMI) (Rudinger et al., 2017). 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 (Zhang et al., 2017). However, machine-generated text might look artificial, and work has shown that text generation has its own set of quality issues (Mathur et al., 2020). 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, as discussed in Subsection 6.2. Additionally, previous work (Roit et al., 2019) 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.

## A.5 DQI Components

DQI shows the (i) the overall data quality and (ii) the impact of new data created on the overall quality. In this paper, higher quality implies lower artifact presence and higher generalization capability. DQI clubs artifacts into seven broad aspects of text, which cover the space of various possible interactions between samples in an NLP dataset. Please refer to (Mishra et al., 2020b) and (Mishra et al., 2022) for a full explanation of parameters.

## A.6 Interface Design Intuitions

**Careful Selection of Visualizations** Prior to the design of test cases and a user interface, data visualizations highlighting the effects of sample addition are built. Considering the complexity of the formulas for the components of empirical DQI, we carefully select visualizations to help illustrate and analyze the effect to which individual text properties are affected.

### A.6.1 Vocabulary

#### Which Characteristics of Data are Visualized?

The contribution of samples to the size of the vocabulary is tracked using a dual axis bar chart. This displays the vocabulary size, along with the vocabulary magnitude, across the train, dev, and test splits for the dataset. By randomizing data splits, the distribution of vocabulary across dataset samples can clearly be identified. We use a dual axis chart as juxtaposition of the vocabulary magnitude against raw counts of words better reflects the evenness of the vocabulary distribution; it is not useful to have only a few samples contributing new words as other samples automatically become easy for the model to solve.

To further clarify the contribution of individual samples to vocabulary, the distribution of sentence lengths is plotted as a histogram. Each sample contributes two sentences, i.e., the premise and hypothesis statements. Figure 10 illustrates this. The histogram provides analysts with a frame of reference to identify gaps or outliers in the distribution, essential for determining which sentences are undesirable for the corpus due to extremely high (low inductive bias) or extremely low (high artifact-spurious bias) vocabulary contribution.

### A.6.2 Inter-sample N-gram Frequency and Relation

#### Which Characteristics of Data are Visualized?

There are different granularities of samples that

are used to calculate the values of this component, namely: words, POS tags, sentences, bigrams, and trigrams. The granularities' respective frequency distributions and standard deviations are utilized for this calculation.

#### Bubble Chart for visualizing the frequency distribution:

A bubble chart is used to visualize the frequency distribution of the respective granularity. This design choice is made in order to clearly view the contribution made by a new sample when added to the existing dataset in terms of different granularities. The bubbles are colored according to the bounds set for frequencies by the hyperparameters, and sized based on the frequency of the elements they represent. Additionally, some insight into variance can be obtained from this chart, by observing the variation in bubble size.

#### Bullet Chart for impact of new sample:

The impact of sample addition on standard deviation can be viewed using the bullet chart. The bullet chart is useful to visually track performance against a target (in this case ideal standard deviation), displaying results in a single column; it looks like a thermometer and is therefore easy to follow. The red-yellow-green color bands for each granularity represent the standard deviation bounds of that granularity. The vertical black line represents the ideal value of the standard deviation of that granularity. The two horizontal bars represent the value of standard deviation before and after the new sample's addition. Figure 12 illustrates the visualization.

### A.6.3 Inter-sample STS

#### Which Characteristics of Data are Visualized?

The main units used in this DQI component are the similarity values between sentences across the dataset. This refers to either premise or hypothesis statements, relative to all other premise/hypothesis statements. In order to understand the similarity relations of sentences, a force layout and horizontal bar chart are used. This is illustrated in Figure 14.

#### Force Layout for Similar Sentence Pairs

In the force layout, those sentence pairs with a similarity value that meets the minimum threshold are connected. Each node represents a sentence. The thickness of the connecting line depends on how close the similarity value is to the threshold. Similarity values are used to create this network, as the aim of this component is to drill down into whether a sample has sufficient inductive bias (i.e., is closely

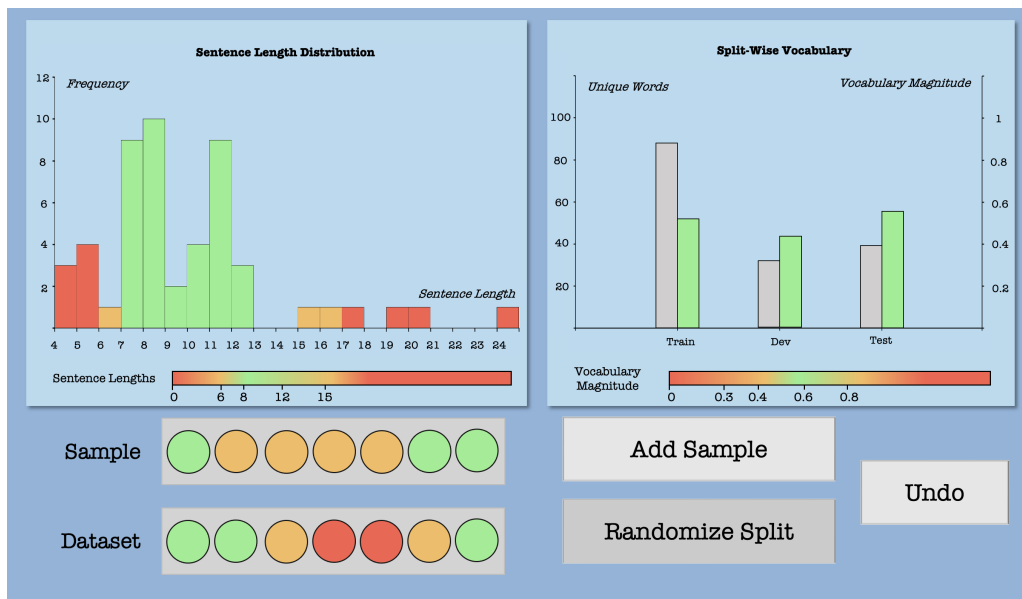


Figure 10:  $DQI_{c1}$  Visualization Prior to New Sample Addition

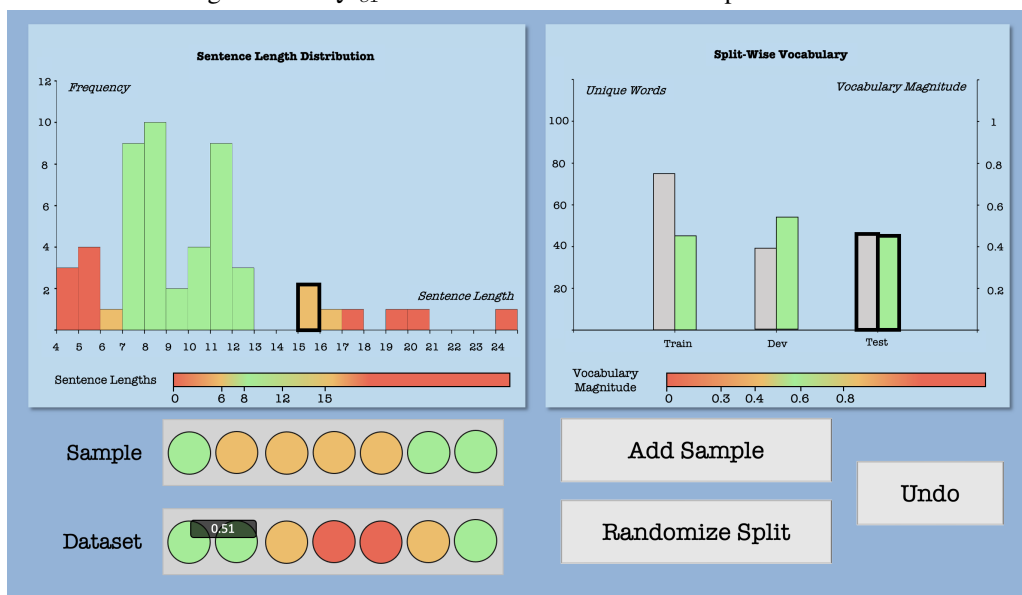


Figure 11:  $DQI_{c1}$  Visualization On New Sample Addition

linked to a sufficient extent to existing samples), and also if a sample is too similar (spurious bias) to existing samples.

**Horizontal Bar Chart for Most Similar Sentences** In the horizontal bar chart, the sentences that are most similar to the given sentence are ordered in terms of their similarity value. The bar colors are centered around the threshold. This helps identify the most important subset to juxtapose the given sample against; the analyst can use this subset to for instance, decide if a moderate quality sample requires a small or big change in order to reach acceptable quality..

#### A.6.4 Intra-sample Word Similarity

##### Which Characteristics of Data are Visualized?

In this section, A sample's word similarity is viewed in terms of premise-only, hypothesis-only, and both. The relationship between non-adjacent words in the sample's sentences is analyzed specifically.

##### Overview Chart for Average Word Similarities and Heatmap for Single Sample

The overview chart that is used is a tree map, which uses the average value of all word similarities per sample- i.e., concatenated premise and hypothesis- to color and group its components. This is illustrated in Figure



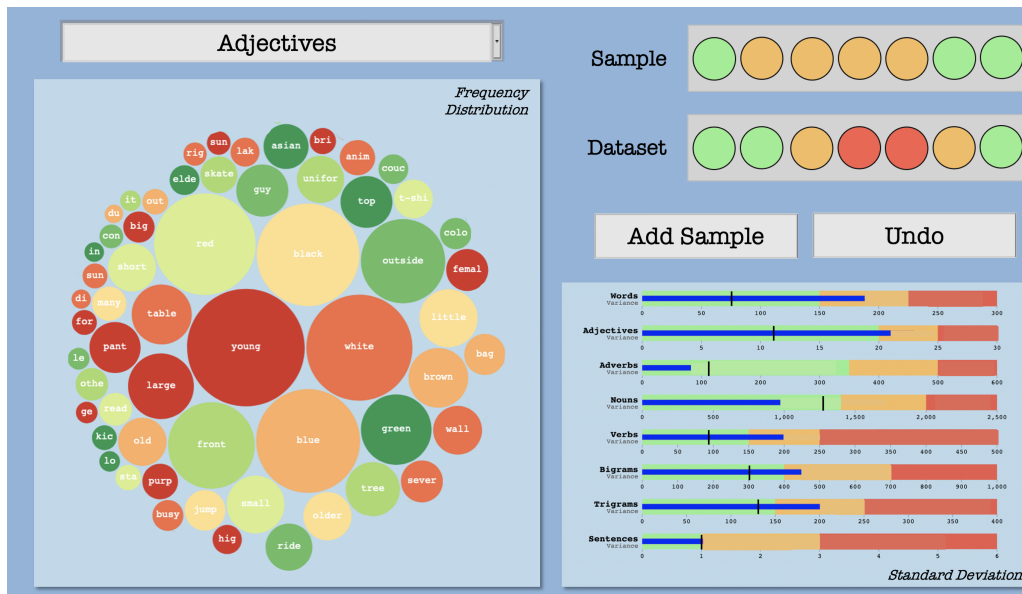


Figure 12:  $DQI_{c2}$  Visualization Prior to New Sample Addition

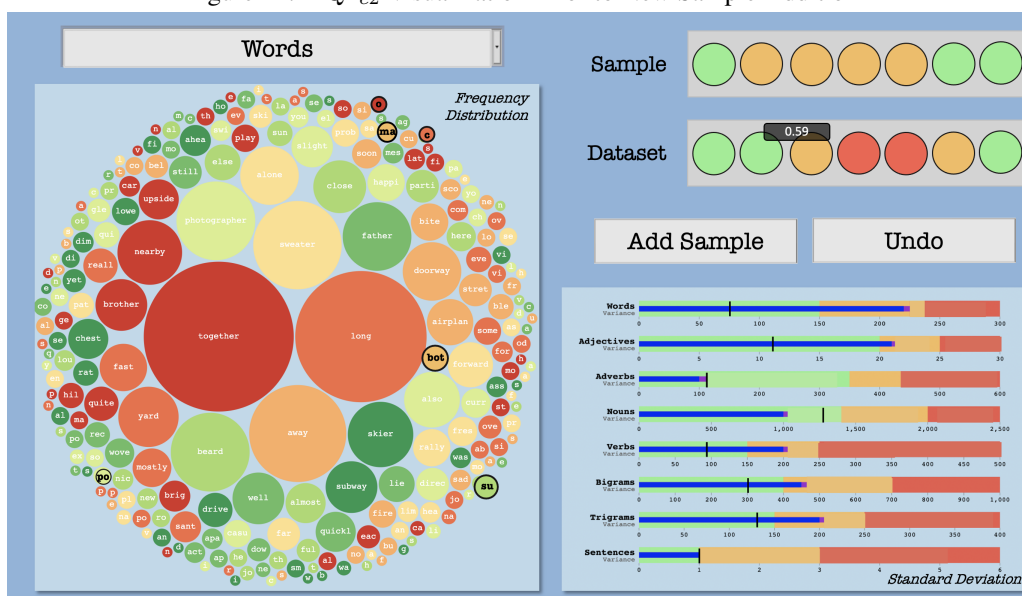


Figure 13:  $DQI_{c2}$  Visualization On New Sample Addition

16. Treemaps capture relative sizes of data categories, allowing for quick perception of the items that are large contributors to each category. This makes them ideal to analyze the inter-relationships between different word pairs across sample, in a concise manner.

The treemap also makes it easy to drill down into the specifics of a particular sample even further. This detailed view is provided in the form of a heatmap. All the words in a single sample, are plotted against each other, as shown in Figure 18. The heatmap provides a mechanism for word-level drill down of sample similarity. Like with the previous component, this helps provide the analyst with a

frame of reference as to whether a moderate quality sample can be sent to TextFooler or not.

### A.6.5 Intra-sample STS

**Which Characteristics of Data are Visualized?** Premise-Hypothesis similarity is analyzed on the basis of length variation, meeting a minimum threshold, and similarity distribution across the dataset. The first is addressed already in the vocabulary property by viewing the sentence length distribution. The other two are visualized using a histogram and kernel density estimation curve, as shown in Figure 19.

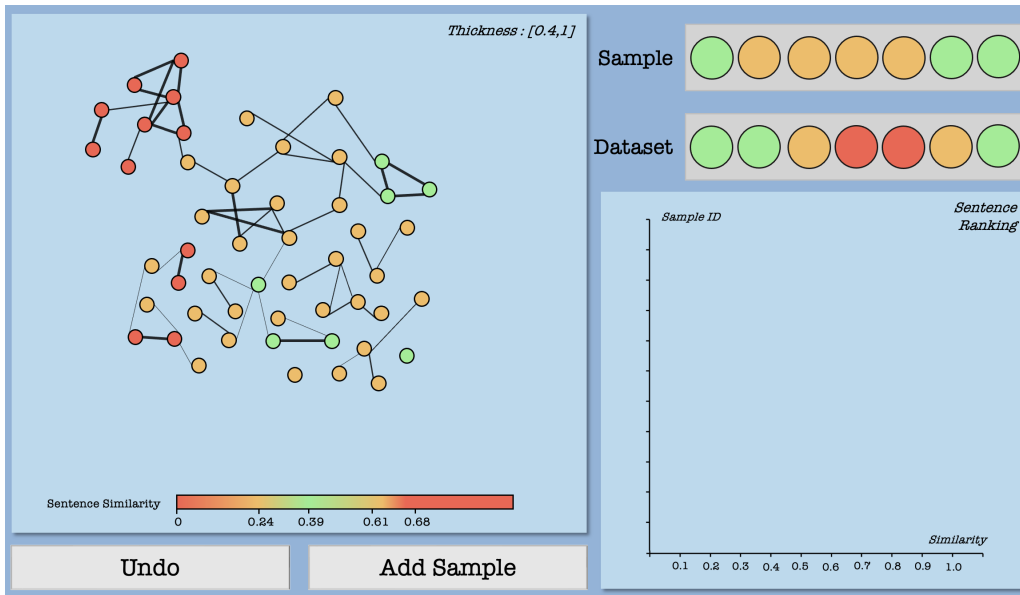


Figure 14:  $DQI_{c3}$  Visualization Prior to New Sample Addition

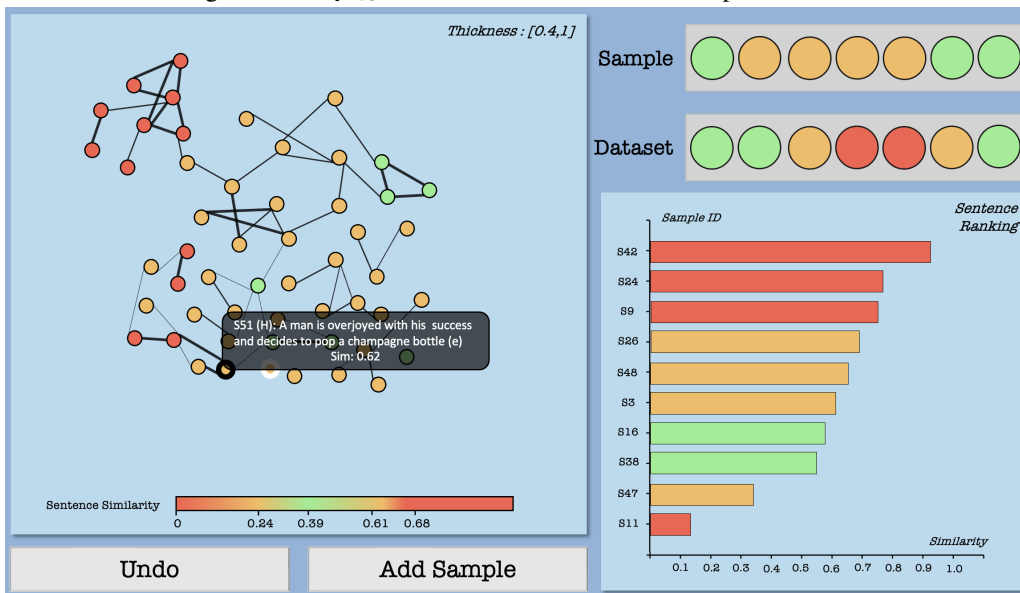


Figure 15:  $DQI_{c3}$  Visualization On New Sample Addition

**Histogram and Kernel Density Curve for Sample Distribution** The histogram represents the distribution of the samples, and is colored by centering around the threshold as the ideal value. The number of bins can be changed, and therefore multi-level analysis can be conducted. This helps identify the proportion of samples that are of non-optimal range.

The kernel density curve is used to check for the overall skew of the distribution. KDE helps visualize the distribution sans user defined bins; this is colored to reflect the bi-linear color scale, and indicates the probability with which a sample will be correctly solved by a model given its similarity.

This helps contextualize the level of artifacts relating to word overlap and sentence similarity in the sample.

#### A.6.6 N-Gram Frequency per Label

##### Which Characteristics of Data are Visualized?

This component drills down on the second component, to view the patterns seen in granularities per label. There are two small multiples charts, divided based on label, used in this view- a violin plot and a box plot.

**Violin plot and Kernel Density Curve for Skew of Distribution:** The violin plots are structured to display both jittered points, according to their

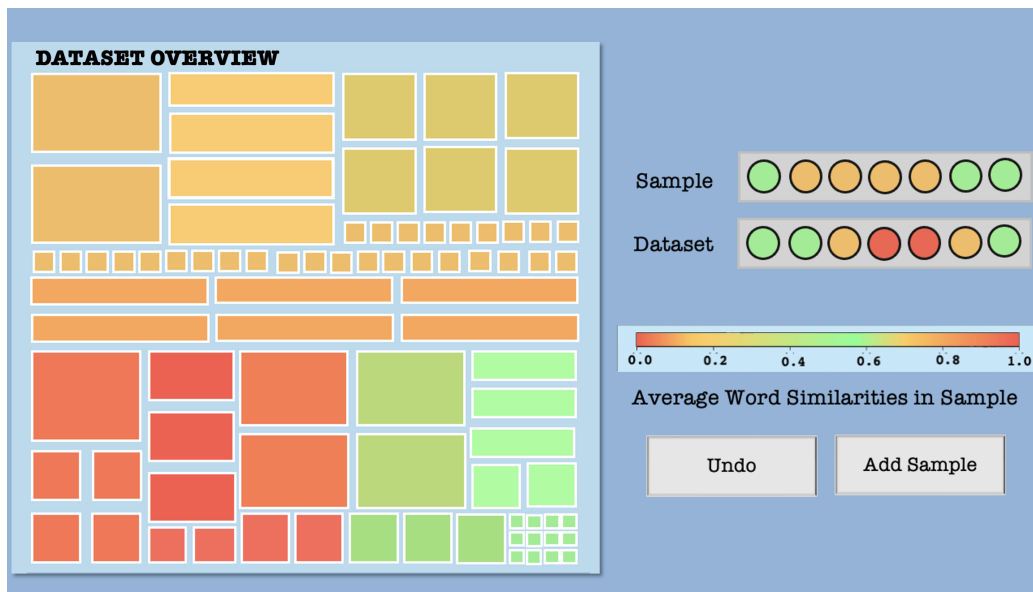


Figure 16:  $DQI_{c4}$  Visualization Prior to New Sample Addition

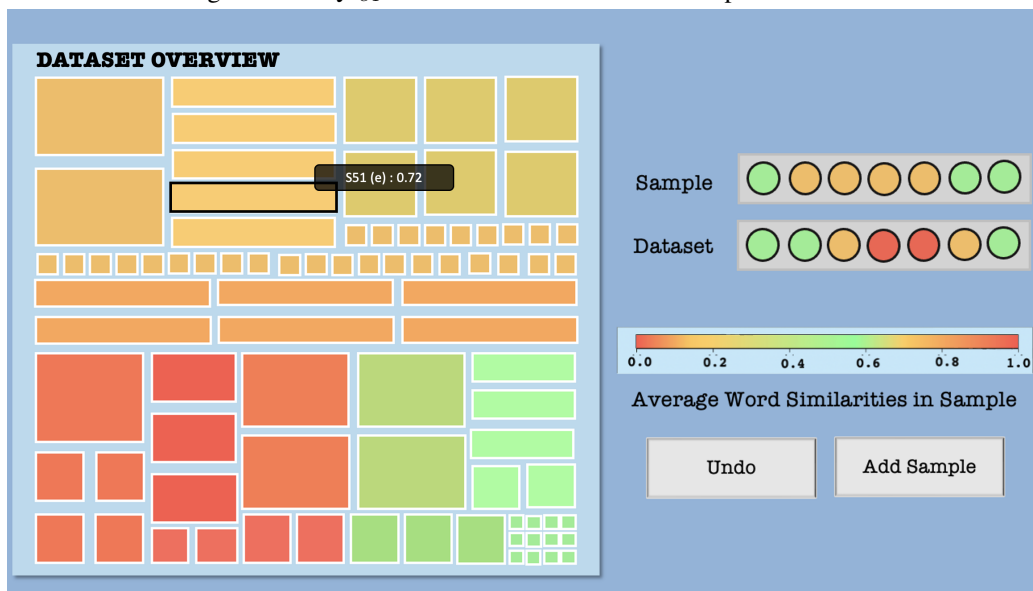


Figure 17:  $DQI_{c4}$  Visualization On New Sample Addition: Dataset View

frequency distribution, as well as a kernel density curve to judge the skew of the distribution. The points each represent an element of the granularity.

**Box Plots for More Information** The box plots are used to garner more information about the distribution, in terms of its min, max, median, mean, and inter quartile range. These help further characterize the distribution, as well as provide a quantitative definition of the skew seen using density curves. Jittered points representing elements are present in this plot as well.

### A.6.7 Inter-split STS

#### Which Characteristics of Data are Visualized?

Train-Test similarity must be kept minimal to prevent data leakage. This component's main feature is finding the train split sample that is most similar to a given test split sample.

#### Parallel Coordinate Graph for Train-Test Similarity:

A subset of test and train samples, all found to have close similarity within their respective splits, and significant similarity across the splits are plotted as a one step parallel coordinate graph, with test samples along one axis, and train samples along the other. This subset is seeded

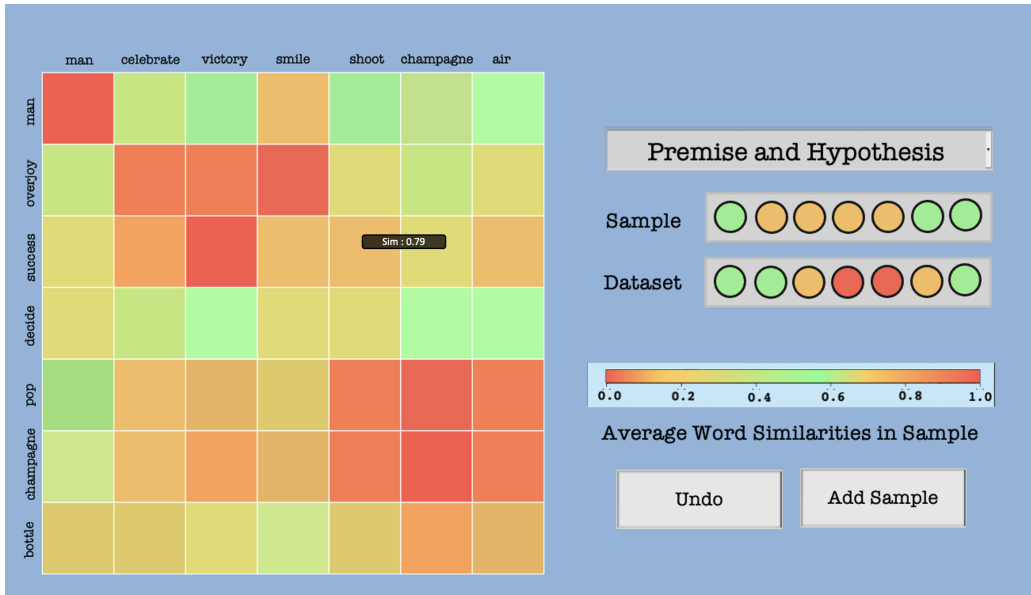


Figure 18:  $DQI_{c4}$  Visualization On New Sample Addition: Sample View

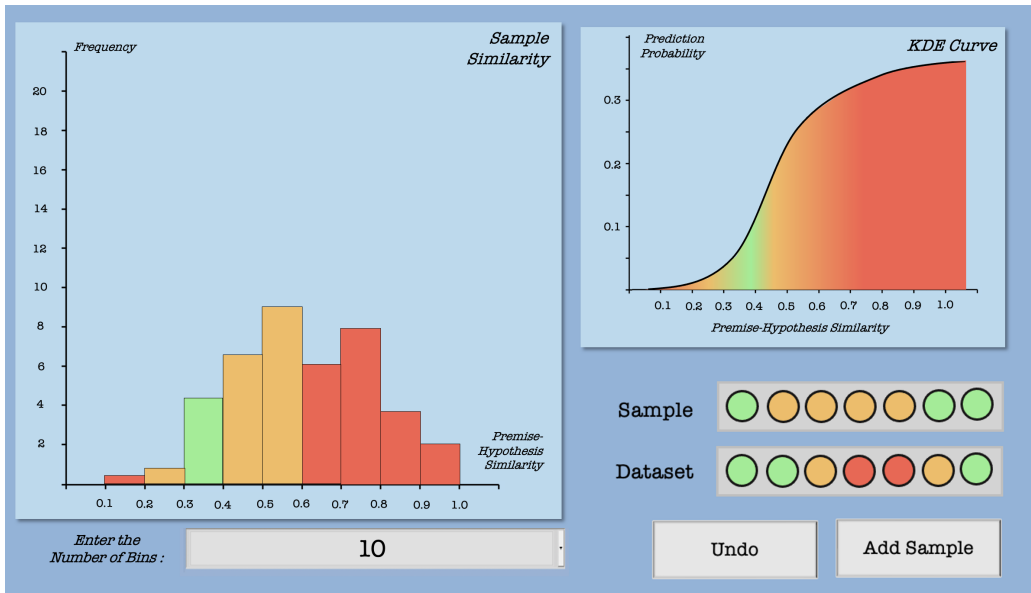


Figure 19:  $DQI_{c5}$  Visualization Prior to New Sample Addition

with those samples closest in similarity to the new sample to be introduced, based on the third component's visualization. The links connecting points on the two axes are drawn between the most similar matches across the split, as shown in Figure 24.

### A.7 AutoFix and TextFooler Examples

See Tables 4, 5.

### A.8 User Study

**AutoFix Suggestions:** See Tables 6, 5.

**NASA TLX:** The NASA Task Load Index (NASA-TLX) is a subjective, multidimensional assessment tool that rates perceived workload in order

to assess a task, system, or team's effectiveness or other aspects of performance (Hart, 2006).

NASA-TLX divides the total workload into six subjective subscales that are represented on a single page. There is a description for each of these subscales that the subject should read before rating. They rate each subscale within a 100-point range, with 5-point steps, as shown in Figure 28. Providing descriptions for each measurement can be found to help participants answer accurately (Schuff et al., 2011). The descriptions are as follows:

- **Mental Demand:** How much mental and perceptual activity was required? Was the

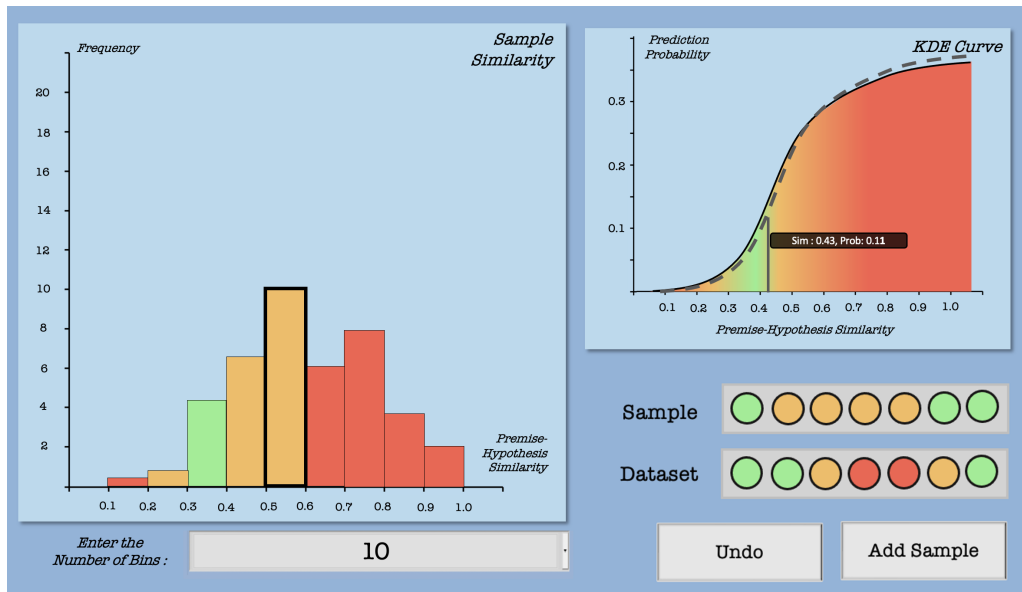


Figure 20:  $DQI_{c5}$  Visualization On New Sample Addition

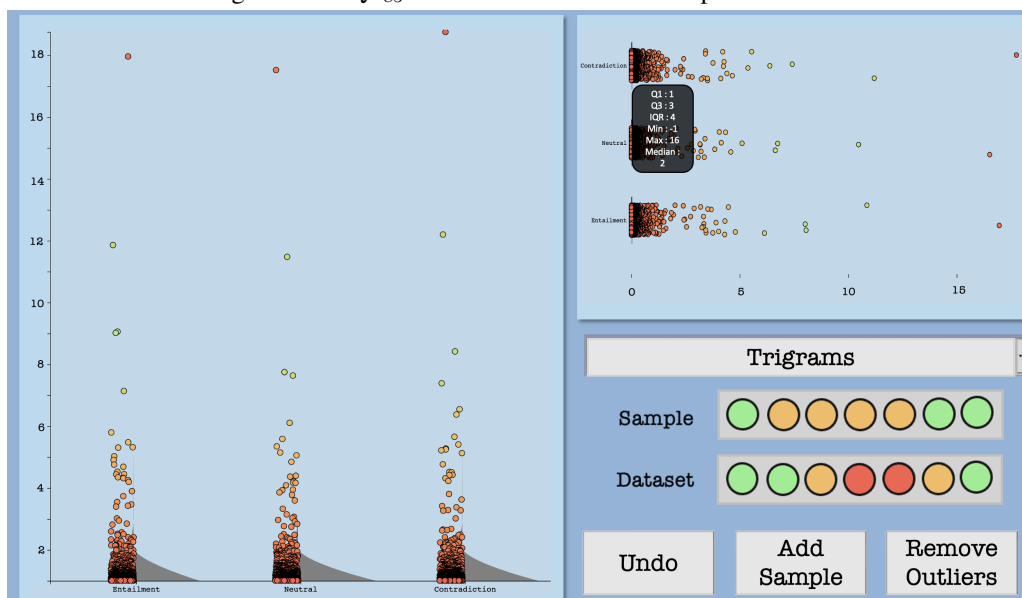


Figure 21:  $DQI_{c6}$  Visualization Prior to New Sample Addition

task easy or demanding, simple or complex?

- **Physical Demand:** How much physical activity was required? Was the task easy or demanding, slack or strenuous?
- **Temporal Demand:** How much time pressure did you feel due to the pace at which the tasks or task elements occurred? Was the pace slow or rapid?
- **Performance:** How successful were you in performing the task? How satisfied were you with your performance?
- **Effort:** How hard did you have to work (men-

tally and physically) to accomplish your level of performance?

- **Frustration:** How irritated, stressed, and annoyed versus content, relaxed, and complacent did you feel during the task?

We record participant demographics— age, gender, and occupation. We also ask participants to rate their familiarity with Visualization and NLP, on a scale of 1 (novice) to 5 (expert). Demographic information is shown in Figure 29. Participants are asked to fill this form at the end of each round of the user study. We also record the number of questions participants successfully create, as well as a

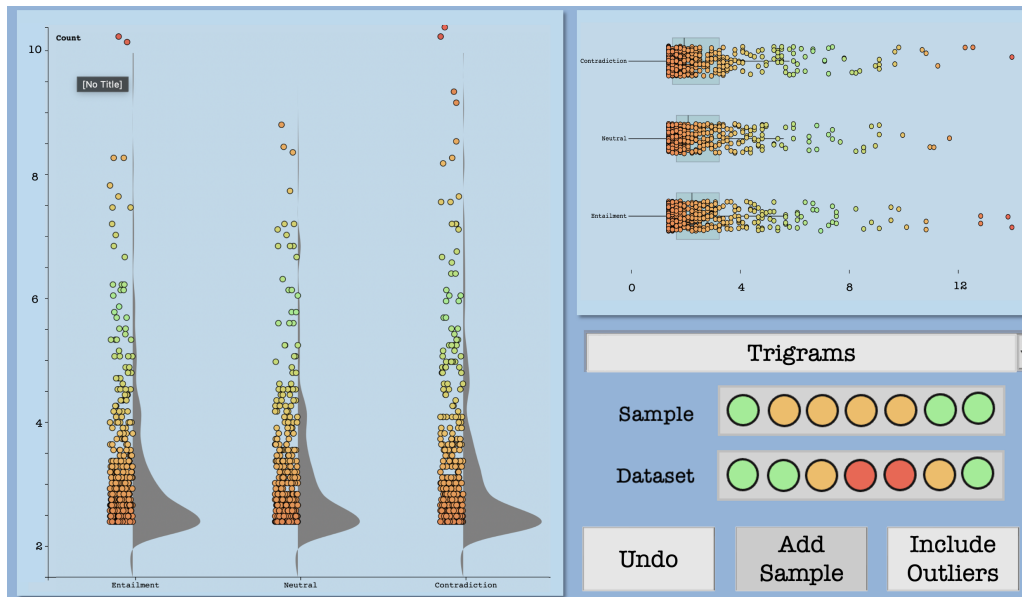


Figure 22:  $DQI_{c6}$  Visualization after removing outliers Prior to New Sample Addition

record of how often participants use each module in the full system round. At the end of the user study, participants are asked what their impression of data quality is, and their free response is recorded.

**Subscale Wise Results:** Individual results of the averaged subscales in Figure 8 are shown in Figures 30,31. Physical demand does not change significantly across user study rounds.

### A.9 Expert and User Comments

**Experts (P):** 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 (P) 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);  $P_2$  said the latter “... enhances readability for understanding the data at different granularities.”.  $P_1$  suggested the inclusion of “... a provenance module within the analyst views to show historical sample edits and overall data quality changes over time to understand how data quality evolves as the benchmark size increases... this would help with the bubble plot and tree map which

will get more cluttered and complex as data size increases”. Additionally  $P_3$  remarked that “The frequency of samples of middling quality should increase as benchmark size increases, but the initial exposure that analysts will have with higher or lower quality samples should lessen the learning curve as they are familiar enough with interface subtleties by the time they begin to encounter more challenging cases.”

**Crowdworkers (C):** When presented with traffic signal feedback, crowdworkers report that the time and effort required to create high quality samples increases—“You need to keep redoing the sample since when you see it’s all red, you know it’s probably not going to be accepted”(C<sub>3</sub>); however, they are more confident about their performance and sample quality “...when there’s green, I know I’ve done it right, and it cuts down on my having to create a lot of samples to get paid” (C<sub>15</sub>). We find that AutoFix usage<sup>7</sup> causes an unexpected increase in mental and temporal demand, as well as frustration; we attribute this to observed user behavior— “I’m not sure how much I trust this recommendation without seeing the colors”(C<sub>12</sub>), and “I’d prefer to change a couple of things since I can’t see the feedback anymore”(C<sub>21</sub>). The drastic improvement over all aspects (highest for frustration) in the case of using the full system is in line with this observation—“This is so easy, I can create samples really fast, and I have a better chance of getting more accepted.”(C<sub>8</sub>) and “Now that I get the feedback along with the recommendation, I can see

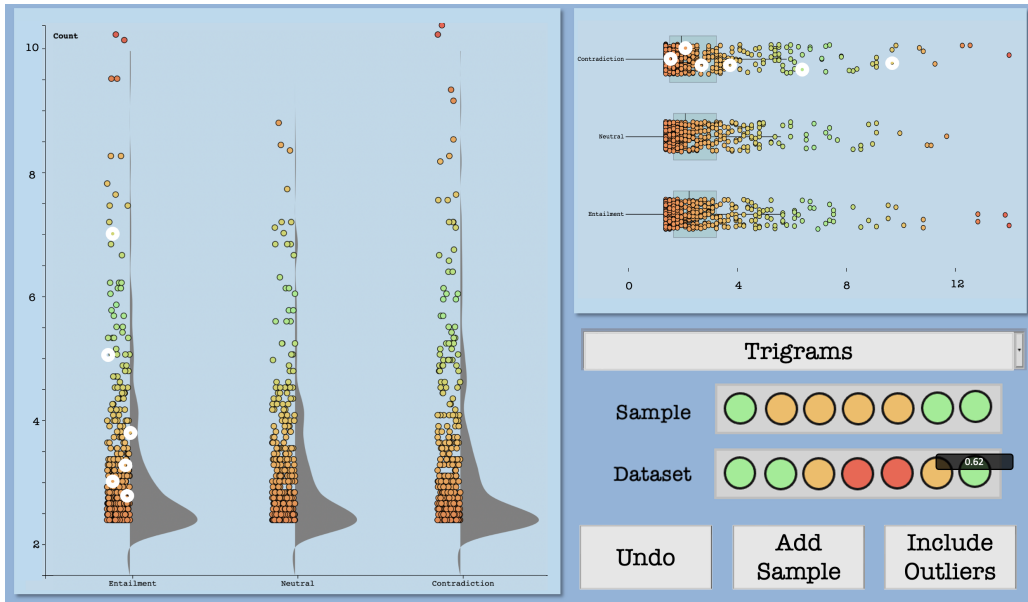


Figure 23:  $DQI_{c6}$  Visualization with mouseover On New Sample Addition

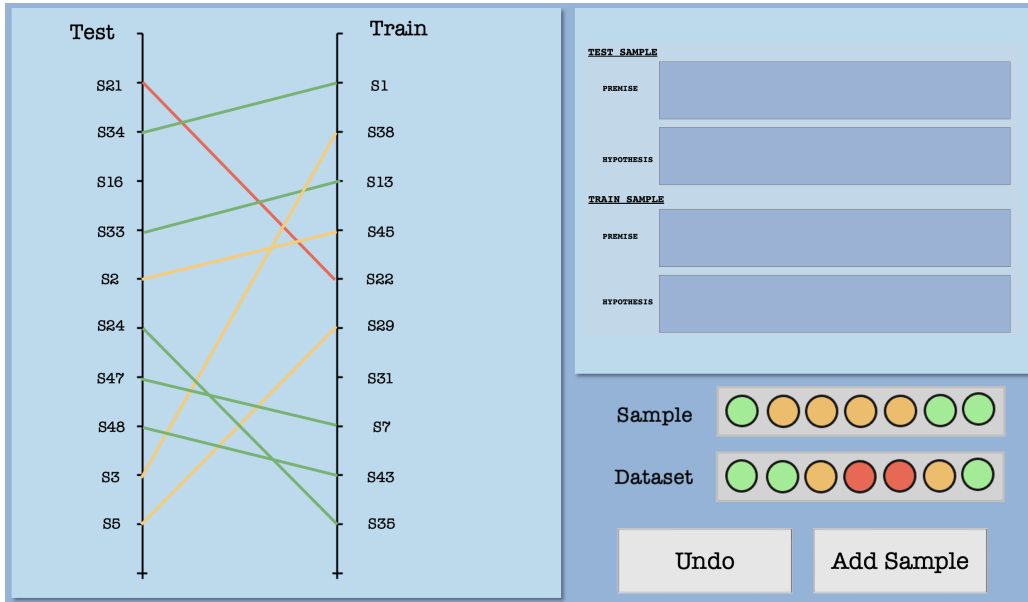


Figure 24:  $DQI_{c7}$  Visualization Prior to New Sample Addition

the quality improvement. So using the recommendation is now definitely faster." ( $C_{12}$ ). The number of questions created per round as well as system scores also follows this trend, across all types of crowdworkers.

**Summary:** Traffic signal feedback initially increases time (+25%) and effort (+60%) required to create high quality samples, as users have to correct them. However they are more confident (performance+ 27%) of sample quality. AutoFix usage causes an unexpected increase in effort (+5%) and frustration (+88.8%), as users do not fully trust recommendations without visual feed-

back. The drastic improvement over all aspects (frustration- 44.4%, mental demand- 38.1%, temporal demand- 29.1%, effort- 20%, average decrease in difficulty- 31.1%, performance+ 34.6%) 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+ 25%, full system+ 83.3%) as well as system scores (traffic signal+ 27.3%, AutoFix+ 13.6%, full system+ 54.5%) also follows this trend, across all types of crowdworkers.

**Analysts (A):** In the case of direct quality feedback, i.e., traffic signals, analysts report an in-

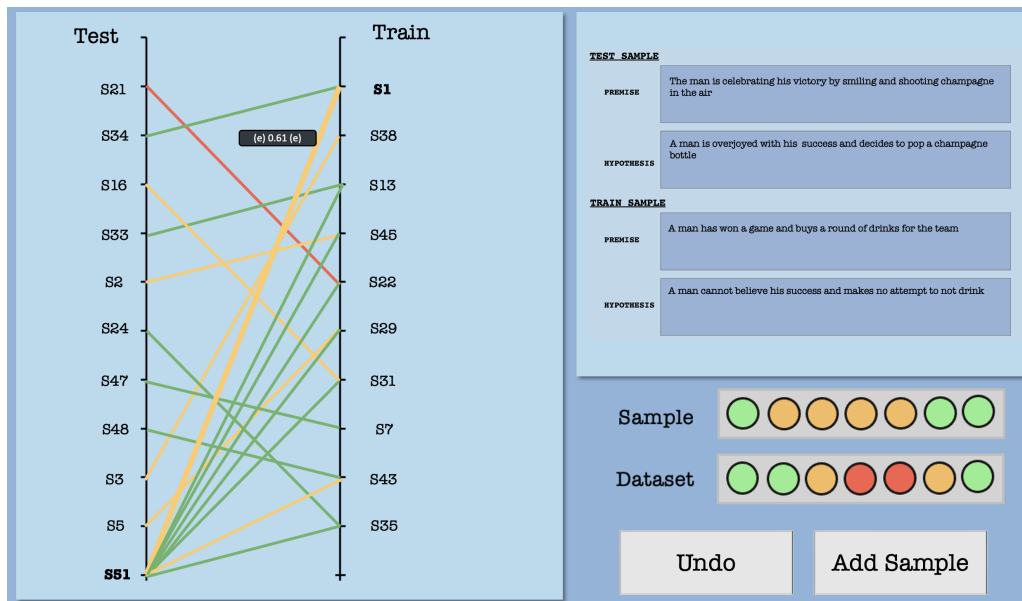


Figure 25:  $DQI_{c7}$  Visualization On New Sample Addition

creased performance and find the task easier—“... it’s easier to directly choose based on quality... and it takes care of typos too, the typo samples are marked down so the work goes pretty fast”(A<sub>3</sub>). 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—“It takes a little time to figure out how to go through the views. I learned that in the samples I looked at, components three and seven seemed to be linked. So I’d look at those first the next time I used the system” (A<sub>6</sub>) and “... it takes me some time to figure out how to read the interfaces effectively, but it does make me more secure in judging sample quality at multiple granularities and that would help if I was doing this for a particular application”(A<sub>1</sub>). Analysts averaged behavior on TextFooler models the conventional approach quite closely, as analysts are seen to have a tendency to either— “... deciding to reject or repair is difficult when you don’t have the sample or dataset feedback... and what if the repaired sample still isn’t good enough?”(A<sub>4</sub>), or— “ I like having this option to repair... I don’t need to waste time on analyzing something that isn’t outright an accept or reject, I can send it to be repaired and come

back to it later”(A<sub>8</sub>). When shown the full system, analysts also report improvement in all aspects, particularly mental demand and performance—“I can be sure of not having to redo things since it’s likely that I will be able to get a low hypothesis baseline using this system”(A<sub>2</sub>, A<sub>1</sub>). The visualization usage also improves— “... I went to component three right off the bat this time, I knew that I could look at the linked components...” (A<sub>6</sub>). Altogether, sample evaluation by analysts increases, following this trend, and analysts are more assured of their performance.

**Summary:** Analysts find the task easier (effort—**-19.3%**, performance—**+26.9%**) with traffic signal feedback, as quality is clearly marked. When analysts are shown the visualization interfaces, they are explicitly taught to differentiate how the traffic signal colors in the visualizations indicate a sample’s effect on the overall dataset quality. Analysts can toggle between the states of original dataset and new sample addition. We find that analysts initially find toggling 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—**+15.9%**). Analysts’ average 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—**-14.3%**), particularly mental demand (**-19.2%**) and performance



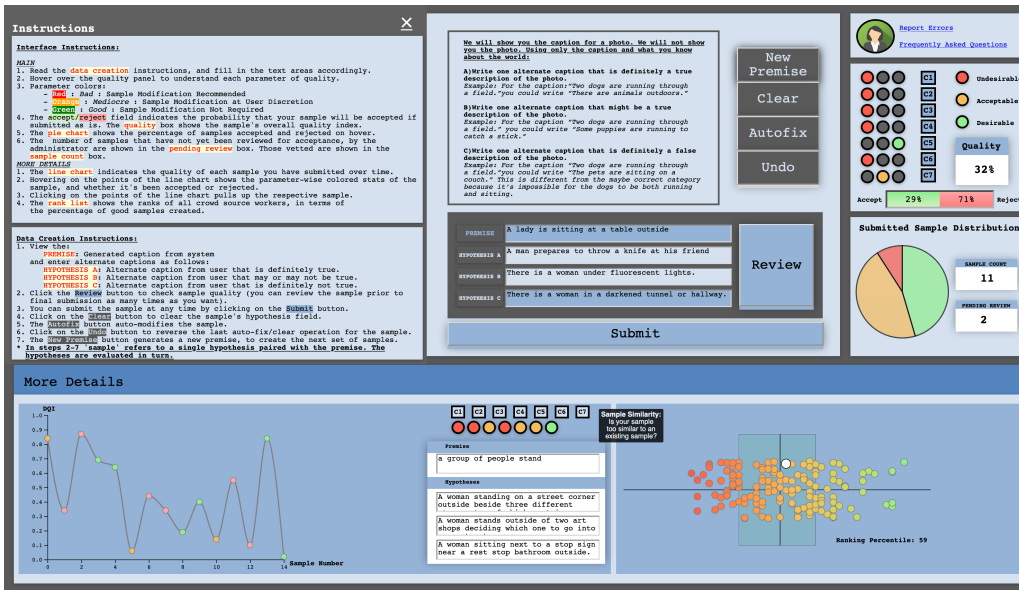


Figure 26: Crowdworquer-View

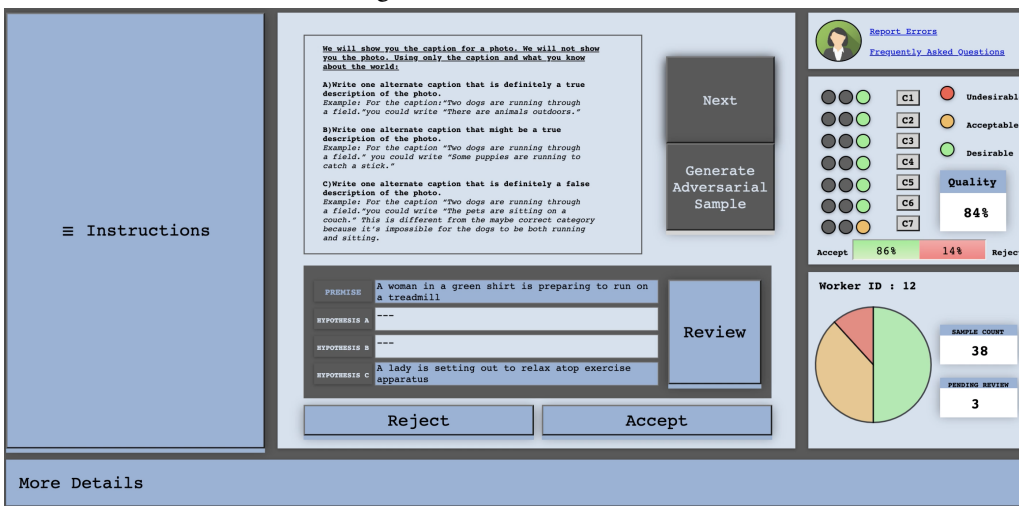


Figure 27: Analyst View

(+30.8%), 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- +94.1%).

Task	Description	Component
New Sample	Adds the sample under review to dataset and updates visualizations.	All
Undo	Removes sample under review from dataset and updates visualizations.	All
Randomize Split	Randomized re-sampling of data across splits in a 70:10:20 ratio.	Vocabulary
Undo Split	Reverses last random split generated.	Vocabulary
Save Split	Freezes split for the remainder of analysis.	Vocabulary
Changing Granularity	View granularity can be changed by selecting drop down option.	Inter-sample N-gram Frequency and Relation, N-Gram Frequency per Label
Change Heat Map View	Using the drop down, the heatmap shows word similarities for the (a) premise, (b) hypothesis, or (c) both sentences.	Intra-sample Word Similarity
Rebinning Histogram	By filling a new value in the textbox, the number of bins in the histogram changes to that value.	Intra-sample STS
Remove Outliers	Removes elements with frequency count less than median count of granularity being viewed.	N-Gram Frequency per Label
Include Samples	Displays all elements for a granularity.	N-Gram Frequency per Label

Table 3: Task Descriptions for Visual Interfaces

Premise	Orig. Hypothesis	DQI	Suggested Words	New Hypothesis based on suggestions	New DQI
A woman, in a green shirt, preparing to run on a treadmill.	A woman is preparing to sleep on a treadmill	2.4650170	preparing,sleep	A woman is organizing to rest on a treadmill	2.5275722
The dog is catching a treat	The cat is not catching a treat	2.752542	catching	the cat is not getting a treat	3.6909140
Three young men are watching a tennis match on a large screen outdoors	Three young men watching a tennis match on a screen outdoors, because their brother is playing	2.6435402 891414217	young,watching, playing	Three youthful men observing a tennis match on a screen outdoors, because their brother is performing.	2.6787982

Table 4: A few samples for Autofix with Intra Sample STS in DQI

Premise	Orig. Hypothesis	DQI	New Hypothesis	New DQI	Label
A woman and a man sweeping the sidewalk.	The couple is sitting down for dinner.	2.416	The couple is meeting for dinner.	3.479	Contradiction
A woman enjoying the breeze of a primitive fan.	The woman has a fan.	2.127	The woman owns a fan.	2.733	Entailment
There is a man in tan lounging outside in a chair.	A man is preparing for vacation.	2.801	A man is arranging to take a vacation.	3.502	Neutral

Table 5: Examples for TextFooler, with DQI’s Intra-sample STS values for existing SNLI samples.

Premise	Orig. Hypothesis	DQI	Suggested Words	New Hypothesis based on suggestions	New DQI
A woman, in a green shirt, preparing to run on a treadmill.	A woman is preparing to sleep on a treadmill	2.4650170	preparing,sleep	A woman is organizing to rest on a treadmill	2.5275722
The dog is catching a treat	The cat is not catching a treat	2.752542	catching	the cat is not getting a treat	3.6909140
Three young men are watching a tennis match on a large screen outdoors	Three young men watching a tennis match on a screen outdoors, because their brother is playing	2.6435402 891414217	young,watching, playing	Three youthful men observing a tennis match on a screen outdoors, because their brother is performing.	2.6787982
A man in a green apron smiles behind a food stand	A man smiles	3.2367785	smiles	A person is grinning.	6.303777

Table 6: A few samples for Autofix with ISSTS in DQI

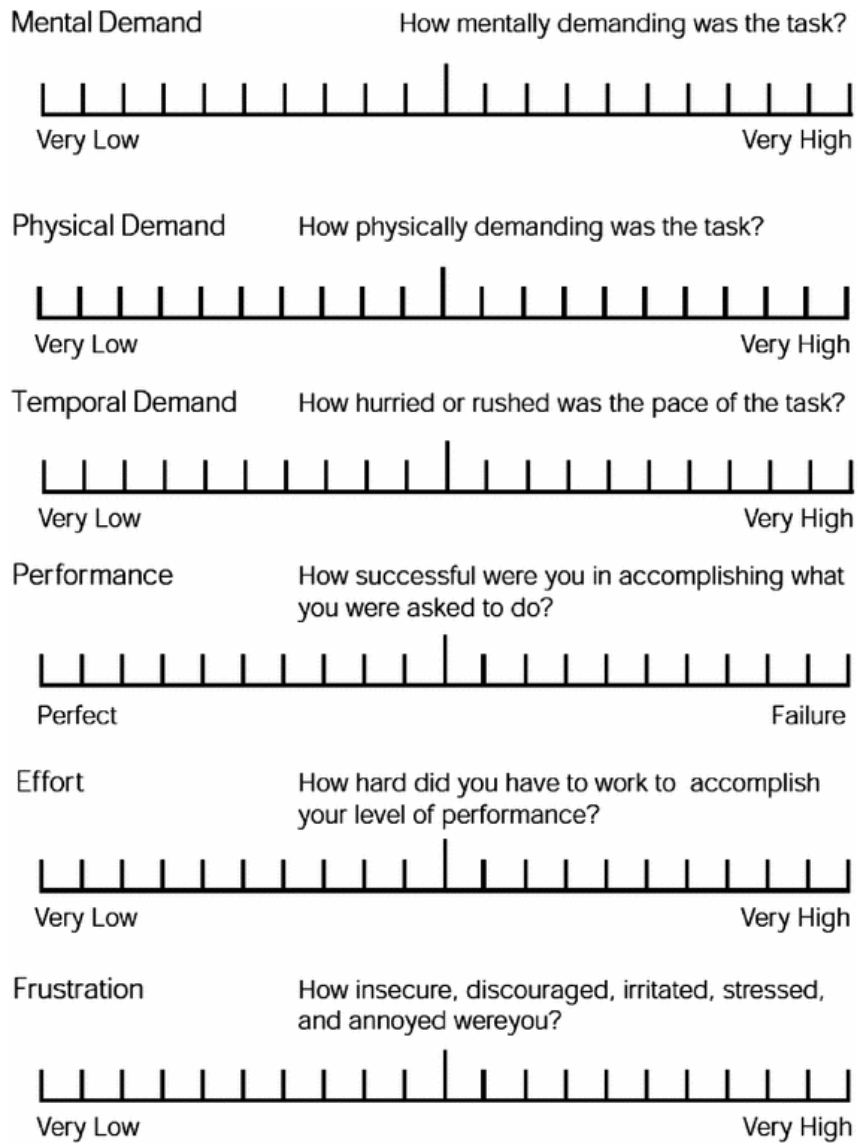


Figure 28: NASA TLX Form

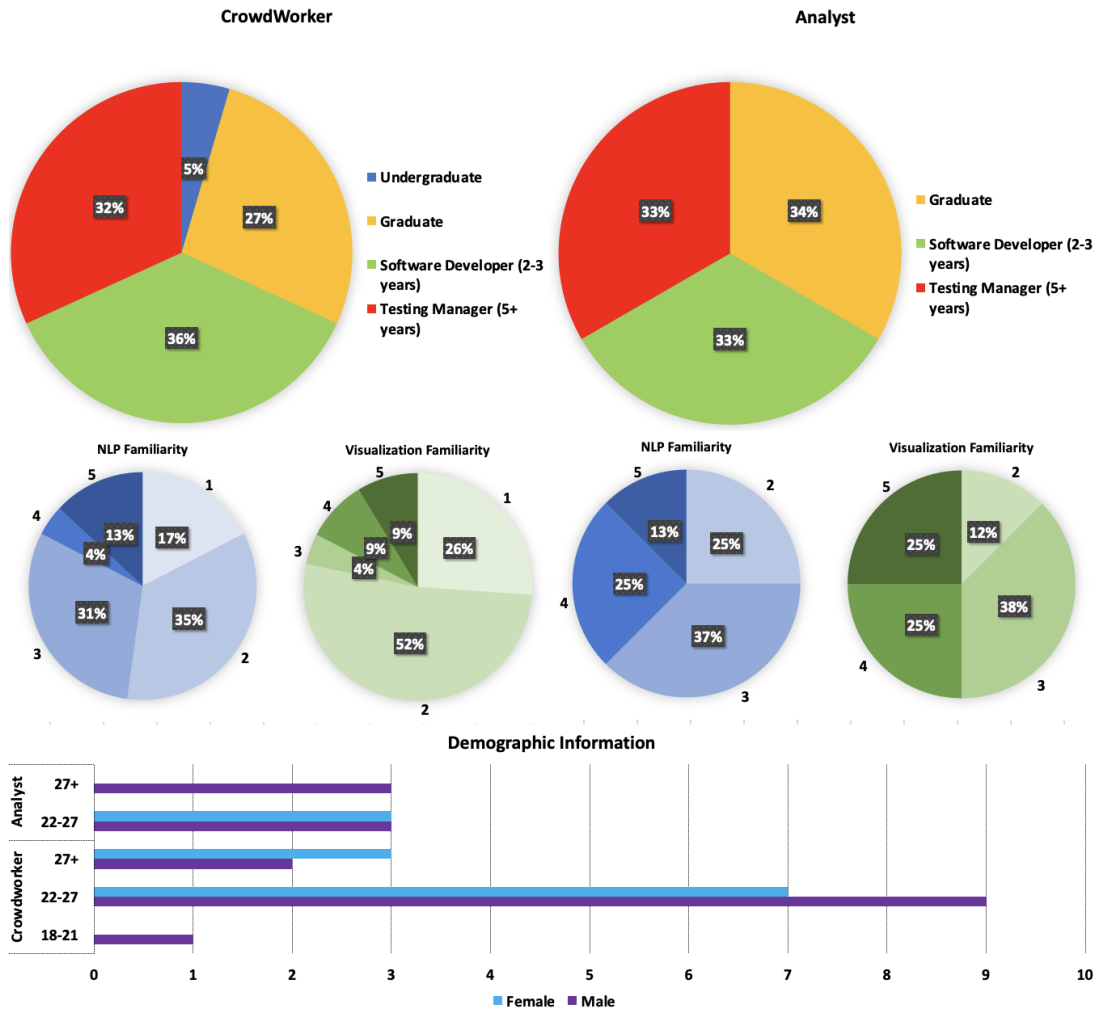


Figure 29: Demographic information for the User Study

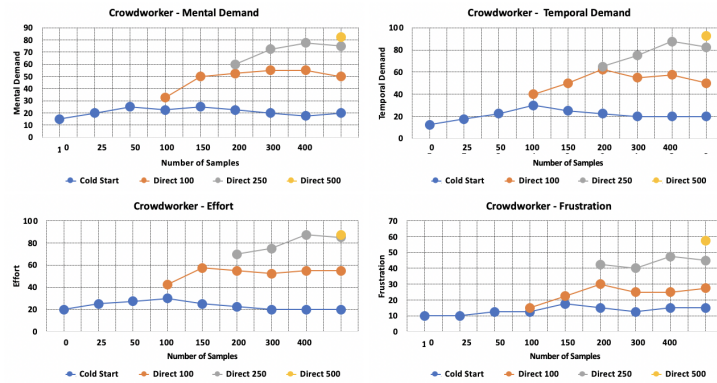


Figure 30: NASA TLX- Crowdworker Subscale Results

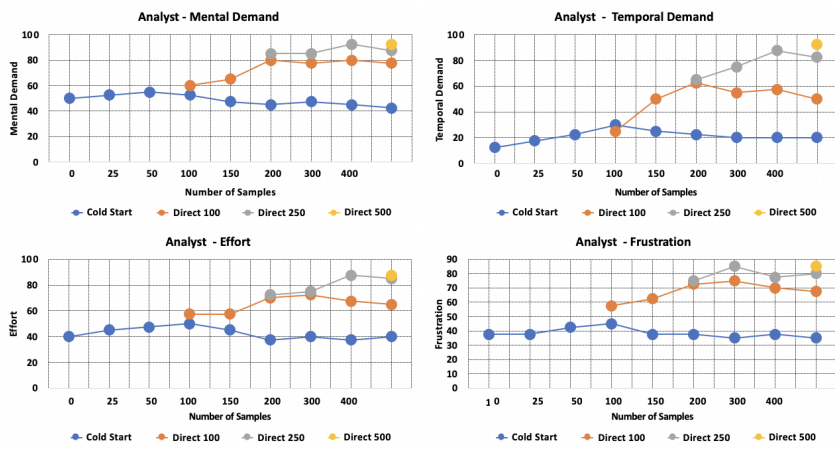


Figure 31: NASA TLX– Analyst Subscale Results