Quality Control for Crowd Workers and for Language Models: A Framework for Free-Text Response Evaluation with No Ground Truth

Online Appendix

APPENDIX A – WORKERS' Q&A DATASETS

Each of our two purposely compiled datasets included 600 responses generated by two mutually exclusive sets of 40 online workers. Each set of workers responded to 15 questions, related to two topics. For each topic the workers were presented a text that included the first few paragraphs of a Wikipedia article. We maintained a word count in each text of 200- to 300 words.[1] In both datasets, the workers were recruited from the Prolific.com platform and were native English speakers.

A.1. First Dataset – Science/Technology- and Sports-Related Questions

For our first dataset, one of the two topics was science and technology; for this topic, participants read a text about the Voyager 1 spacecraft, based on the Wikipedia page https://en.wikipedia.org/wiki/Voyager_1. They subsequently responded to 8 questions about the text. The other topic was sports; for this topic, participants read a text concerning the Iron Man Triathlon sporting event, based on the Wikipedia page: https://en.wikipedia.org/wiki/Ironman_Triathlon, and subsequently responded to 7 questions.

Each response was independently scored for correctness (from 0 to 100%) by two evaluators (two of the authors). Before scoring the responses, the evaluators determined and agreed upon a correct answer for each question, and then used the following general scoring scheme:

$$Evaluation\ Score_{\mathrm{ij}} = \frac{|topics_{correct,j} \cap topics_{ij}|}{|topics_{correct,j} \cup topics_{ij}|}.$$

We note that the evaluators had discretion in penalizing responses that mentioned correct notions in an incomplete or ambiguous manner. After the evaluators independently scored each response, in cases where evaluators' scores differed significantly (difference >33%; ~10% of the cases), the evaluators discussed and

[1] We deleted several sentences from each text to maintain a 200- to 300-word limit per topic.

reconciled the differences. Following this procedure, the average Pearson correlation for the two evaluators' scores, across questions, was 0.94.

Table A1 presents a list of the questions and statistics regarding the responses and their evaluation. One informative statistic is the average score per question. As observed, the average score per question ranged from 36% to 100%. Importantly, 6 of the 15 questions received an average score ≤50%. This result indicates that 40% of the questions were rather challenging for the workers. Such low accuracy levels may seemingly introduce a challenge for voting schemes. Yet, the AQER framework, which includes the iterative reweighting component, was able to overcome this challenge and produce accurate evaluations.

Table A1. Science/Technology and Sports - Questions and Response Statistics

Topic	Question	Number of responders	Average score	Correlation between a worker's score for her/his response to a given question, and the worker's average score of her/his responses to the remaining questions
Voyager 1	With which communication network does Voyager 1 communicate to receive routine commands and to transmit data to Earth?	40	50.0%	0.57
	How did the voyager team test Voyager in late 2017?	40	48.1%	0.62
	Information gathered from which spacecraft helped Voyager's engineers design Voyager to cope more effectively with the intense radiation environment around Jupiter?	40	46.6%	0.57
	Next to which planets did Voyager 1 made flybys?	40	74.1%	0.41
	Which inexpensive improvement was utilized in Voyager 1 to enhance radiation shielding?	40	81.2%	0.21
	Why did NASA prioritize a flyby next to Titan rather than next to Pluto?	40	73.4%	0.60
	Why is Voyager 1 not expected to operate after 2025?	40	67.3%	0.65
	Which aspects of the planets it visited, did Voyager 1 study?	40	35.8%	0.48
Iron Man Triathlon	Which fields of sport are included in an Ironman Triathlon?	40	100.0% (All responses were correct)	n/a
	What was the question that the founders of the race were trying to settle by establishing the Ironman competition?	40	79.9%	0.26
	Why was it decided to shave 3 miles off the bike course?	40	48.0%	0.45
	Which phrase was written on the last page of the rule sheet that athletes received before the first Ironman race?	40	73.5%	0.39
	Why was Hawaii a natural choice for the site of the first Ironman race?	40	44.7%	0.27
	Why do different Ironman races have different time limits?	40	58.6%	0.26
	Why did John Collins mention the Belgian cyclist Eddy Merckx?	40	70.9%	0.25

Another informative statistic relates to workers' inherent capabilities, measured as the correlation between worker $W_i's$ score on a specific question Q_j and her/his average score across all other questions: $\rho\left(Evaluation\ Score_{\mathrm{i,j}}, \frac{\sum_{l\neq j} Evaluation\ Score_{\mathrm{lj}}}{n-1}\right)$. In our experiment, the correlation values ranged from 0.21 to 0.65. The positive correlation values indicate that workers show inherent levels of correctness in their responses and that workers' performance is consistent, to some extent, across responses.

A.2. Second Dataset – History- and Movie-Related Questions

For the second dataset, the first topic was history; specifically, participants read a text about the historical landing at Normandy during World War II, based on the Wikipedia page: https://en.wikipedia.org/wiki/Normandy_landings. They subsequently responded to 8 questions. The second topic was movies; in this case, participants read a text relating to the movie *The Wonderful World of the Brothers Grimm*, based on the Wikipedia page: https://en.wikipedia.org/wiki/The_Wonderful_World_of_the_Brothers_Grimm. They subsequently responded to 7 questions.

As in the first dataset, each response was independently scored for correctness (from 0 to 100%) by two evaluators (an author and a graduate student). In cases where evaluators' scores differed significantly (difference >33%; ~10.1% of the cases), the evaluators discussed and reconciled the differences. Following this procedure, the average Pearson correlation for the two evaluators' scores, across questions, was 0.936.

Table A2 presents a list of the questions and statistics regarding the responses and their evaluations. In this case, the average score per question ranged from 34% to 95.9%, and 3 of the 15 questions received an average score ≤50%—indicating that 20% of the questions were rather challenging for the workers. Again, as in the first dataset, AQER's iterative reweighting component enabled the framework to overcome this challenge and produce accurate evaluations.

Regarding workers' inherent capabilities—measured, as in the first dataset, as

$\rho\left(Evaluation\ Score_{\mathrm{ij}}, \frac{\sum_{l\neq j} Evaluation\ Score_{\mathrm{lj}}}{n-1}\right)$ — we obtained correlation values ranging from 0.1 to 0.61. These positive values suggest that here, too, workers show inherent levels of correctness in their responses and that workers' performance is consistent, to some extent, across responses.

Table A2. History- and Movie-Related Questions – Questions and Response Statistics

Topic	Question	Number of responders	Average score	Correlation between a worker's score for her/his response to a given question, and the worker's average score of her/his responses to the remaining questions
Normandy Landings	What was the purpose of Operation Bodyguard?	40	63.7%	0.61
	Why was the landing delayed?	40	91.7%	0.21
	What were the names of the sectors that the beach was divided into?	40	69.1%	0.42
	After the first delay, why would a further postponement of the invasion mean a delay of at least two weeks?	40	65.3%	0.49
	What obstacles did the German military lay on the beaches?	40	49.6%	0.57
	Why were the casualties heaviest at Omaha Beach?	40	58.4%	0.43
	What did the wind cause during the landing?	40	52.8%	0.32
	What kind of attacks did the Allies conduct before the seaborne invasion?	40	34.0%	0.26
The Wonderful World of the Brothers Grimm	What genre(s) does the film "The Wonderful World of the Brothers Grimm" belong to?	40	52.5%	0.12
	In addition to being a director, what was George Pal's role(s) in the movie?	40	41.8%	0.57
	According to the movie, what type of stories did Wilhelm collect?	40	95.9%	0.19
	According to the movie, how did Wilhelm collect stories?	40	70.1%	0.41
	According to the movie, what happened to the Duke's family history manuscript?	40	83.4%	0.10
	How did Wilhelm become ill with pneumonia?	40	80.8%	0.20
	According to the movie, in Wilhelm's dream, what did the fairy tale characters want?	40	68.7%	0.23

APPENDIX B –IMPLEMENTATION ASPECTS OF THE AQER FRAMEWORK

The AQER framework is modular and can support a host of implementation options. In section B1 we discuss an alternative method to conduct multidimensional voting-based initialization. In section B2 we evaluate the robustness of the AQER framework in case of additional implementation alternatives.

B1 – Initialization using an Entailment-based Voting Scheme

As discussed in the main body of the paper, the first component of the AQER framework is a voting scheme which is used to determine the initial SEA for each question by considering $agg(\{text_{1j} \dots text_{Mj}\})$ where agg is typically the average vote function. Following the calculation of SEA_j, AQER then produces initial scores for each worker by the average cosine similarity between $text_{ij}$ and SEA_j. Indeed, cosine similarity is a popular metric for measuring contextual similarity (e.g., Cer et al., 2018; Gao et al., 2021; Reimers et al., 2019). However, recent research proposes an alternative approach to compare texts, based on the similarity of the texts' content, as measured by context entailment (Androutsopoulos and Malakasiotis, 2010).

In this section, we discuss how entailment can be used as an alternative to cosine similarity and average voting when initializing the AQER framework. Following Androutsopoulos and Malakasiotis (2010) entailment between two responses, $Entailment(R_{lj}, R_{ij})$, is a one-way score that determines the extent to which the second response, R_{ij} entails from the first response R_{lj}. The score provided by the AQER framework for response R_{ij}, is the average of the entailment scores (votes) given by all other responses:

$\hat{s}_{ij} = \frac{1}{M-1}\sum_{l \neq i} Entailment(R_{lj}, R_{ij})$.

Given the asymmetric scores of each response, the initial SEA of each question Q_j is estimated as a weighted multidimensional vote: $SEA_j = \sum_i \frac{\hat{s}_{ij}}{\sum_i \hat{s}_{ij}} \times text_{ij}$.

Specifically, in our empirical evaluations below, $Entailment(R_{lj}, R_{ij})$, was computed via an LLM-based entailment classifier (Lewis et al., 2019)[2] where the first response, R_{lj}, serves as the observation, and the second response, R_{ij} serves as the potential label. However, we note that given the abundant literature on textual Entailment, it is possible to use other classifiers/models to determine the levels of entailment.

[2] See https://huggingface.co/facebook/bart-large-mnli (entailment task, using the zero-shot pipeline)

B2 – Robustness Evaluation

To test the robustness of the AQER framework we experimented with different implementations of AQER. Specifically, in this Appendix, we provide the worker evaluation results for the following implementations of AQER:

a. Using different word embedding such as words embedding based on MPNet (Song et al., 2020), or GPT (using version 3.0 API[3]) as an alternative to our default RoBERTa-based embedding.
b. Replacing the equal weight initialization with a noisy weight initialization. (each worker's initial weight is perturbed by a random noise of up to +/- 10%)
c. Replacing the cosine similarity metric with an alternative measure: negative Euclidean distance. I.e., $\hat{s}_{ij} = -\text{Euclidean distance}(\overrightarrow{text}_{ij}, \overrightarrow{SEA}_j)$.
d. Replacing the way $\hat{s}_{ij}$ and SEA_j are calculated by using the entailment-based approach discussed in Appendix B1.

Table B1 presents the results for these different implementations of AQER. It also reports the results for the best baseline approach (either Li and Fukumoto 2019, or "LLM-as-a-judge", depending on the dataset) and the standard implementation of AQER (as described in the main body of the paper) as reference points. As observed in Table B1, all the different variants of AQER display robust performance – regularly surpassing benchmark approaches or, at minimum, producing comparable results to the best-performing baseline. It is also possible to observe that AQER with Entailment-based initialization, as well as AQER with noisy initialization, converge to the same results (3 digits after the decimal point) as the standard AQER implementation, thus indicating the robustness of the convergence of the iterative procedure and its ability to produce meaningful results. Additionally, we see that the AQER implementation used GPT 3.0-based embedding was very competitive and produced the best results for the CS dataset (outperforming the original AQER implementation).

[3] See text-embedding-3-large in https://platform.openai.com/docs/guides/embeddings

Table B1. Worker Evaluation

Dataset	Standard AQER	AQER with MPNet-based Embedding	AQER with Entailment-based Initialization	AQER with GPT 3 - based Embedding	AQER with Negative Euclidean Distance	AQER with Noisy Weight Initialization	Best Baseline Approach
Science & Sports	**0.950**	0.921	**0.950**	0.933	0.930	**0.950**	0.901
History & Movies	**0.915**	0.887	**0.915**	0.892	0.828	**0.915**	0.793
Computer Science	0.964	0.958	0.964	**0.976**	0.958	0.964	0.960

This table presents the Pearson correlation coefficients between each worker's average response grade as determined by human expert evaluators and the worker's grade calculated using our AQER framework (different variants) and the best baseline approach. The reported results for the semi-synthetic Computer Science dataset represent the average of 25 simulation repetitions. Best AQER variant for each dataset is highlighted in Bold.

B3 – Proof of Convergence of the AQER framework

Without loss of generality, we focus on a specific query Q_j, and on the SEA_j vector associated with the answers R_{ij} (and their corresponding embeddings $text_{ij}$) of all workers W_i to this query. Let $cosine(text_{ij}, text_{true,j})$ denote the cosine similarity between worker W_i's response for Q_j and the true embedding $text_{true,j}$. For brevity, we may use $cosine_{ij}$.

We define d_j as the difference between the true vector of Q_j (i.e., $text_{true,j}$) and the SEA_j vector, computed as: $d_j = error(SEA_j) = 1 - cosine(SEA_j, text_{true,j})$. We denote the iteration index of the AQER process by $\{t_0, t_1, \dots\}$. We assert that, in expectation, if $t_p > t_r$, then:

$$(1)\ d_j^{t_p} \leq d_j^{t_r}$$

AQER's iterative process ensures that, in expectation, the difference between the synthetic exemplary answer vector SEA_j and the true answer $text_{true,j}$ vector decreases with the number of iterations. While individual iterations may introduce temporary deviations, the overall trend exhibits a contraction pattern, ensuring convergence as the process progresses.

Justifying the assertion

According to step 2 of the AQER algorithm,

$$SEA_j^{(t)} = \sum_i weight_i^{(t)} \times text_{ij}.$$

Initially, at *t=0*

$$SEA_j^{(0)} = \sum_i \frac{\hat{s}_{ij}^{(0)}}{\sum_i \hat{s}_{ij}^{(0)}} \times text_{ij}.$$

We want to show that applying updated weights at later iterations leads to a decrease in d_j. Recall that the weights are normalized. Additionally recall that they are computed based on the grades of the workers. That is:

$$weight_i^{(t)} = \hat{s}_i^{(t)} / \sum_i \hat{s}_i^{(t)}$$

Since the grade of each worker is influenced by the quality of the worker's answers, a weighting mechanism systematically reduces the influence of workers whose embeddings $text_{ij}$ deviate significantly from the SEA_j vector, and vice versa. As a result, over time, the SEA_j vector in the AQER process will include smaller portions of the lower-grade workers' answers, and larger portions of the higher-grade workers' answers.

Consequently, the influence of low-quality answers on SEA_j is reduced, while the contribution of high-quality answers – closer to the true answers – is increased. This process ensures that SEA_j moves closer to $text_{true,j}$, reducing d_j over iterations.

The unbiasedness and independence assumptions made imply the following:

The SEA_j vector is, on average, closer to the $text_{true,j}$ vector than the $text_{ij}$ vector. This does not necessarily hold for every individual iteration, as some worker responses may be closer to the true answer than the current SEA_j. However, in expectation, the iterative weighting process ensures that SEA_j moves progressively closer to $text_{true,j}$ over multiple iterations.

Since the AQER process downweights low-quality responses and assigns higher influence to high-quality responses, the overall trend remains **monotonic improvement in expectation**. Even if some individual iterations introduce temporary deviations, the **expected contraction property** ensures that the process follows a **Cauchy sequence**, leading to convergence.

This holds in particular for "bad" workers, whose $text_{ij}$ vectors are, on average, farther from $text_{true,j}$ compared with the $text_{ij}$ vectors of "good" workers. By downweighting "bad" workers' influence in subsequent iterations, we reduce their impact on the next SEA_j vector, effectively decreasing d_j.

As the iterative process continues, the influence of "bad" workers diminishes, leading to an expected reduction in d_j over iterations. While some individual iterations may not strictly reduce d_j, the overall expected behavior is a progressive reduction in deviation.

Cauchyness

To prove convergence of the sequence of the d_j values, we show that the sequence is *Cauchy* (Fridy, 1985), which leads to the sought conclusion.

To show that the sequence $\{d_j^t\}$ is *Cauchy***,** where each element in the sequence is computed based on the previous one, we need to prove that for every $\epsilon>0$, there exists an N such that for all $m,n \geq N$, the distance between the elements satisfies:

$$| d_j^{(n)} - d_j^{(m)} | < \epsilon.$$

We seek a recurrence relation according to which

$$d_j^{(n+1)} = d_j^{(n)} \cdot w_n, \quad 0<w_n<1.$$

We can express the general element as:

$$d_j^{(n)} = d_j^{(0)} \cdot (w_1 \cdot \cdot \cdot w_n),\ 0<w_i<1.$$

Denoting $w_{max} = max(w_i)$, we obtain:

$$d_j^{(n)} <= d_j^{(0)} \cdot w_{max}^{\ n}$$

where $d_j^{(0)}$ is the initial value of the sequence.

Since the iterative process progressively reduces the influence of high-error responses, we expect $d_j^{(n)}$ to follow a contraction pattern. Our goal is to show that this contraction ensures a *Cauchy* sequence, thus proving convergence.

To establish Cauchyness, consider the difference:

$$| d_j^{(n)} - d_j^{(m)} | = | d_j^{(0)} | \cdot | (w_1 \cdot \cdot \cdot w_n) - (w_1 \cdot \cdot \cdot w_m) |$$

W.L.O.G., assume $n<m$, so we can write

$$| d_j^{(n)} - d_j^{(m)} | = | d_j^{(0)} | \cdot | (w_1 \cdots w_n) - (w_1 \cdots w_n \cdot w_{n+1} \cdots w_m) | \text{ , or}$$

$$| d_j^{(n)} - d_j^{(m)} | = | d_j^{(0)} | \cdot | (w_1 \cdots w_n) | \cdot | 1 - (w_{n+1} \cdots w_m) |$$

Since $0<w_i<1$, as $m-n \to \infty$ we can choose N sufficiently large such that:

$$| (w_1 \cdots w_n) - (w_1 \cdots w_n \cdot w_{n+1} \cdots w_m) | < \frac{\epsilon}{| dj(0) |}$$

Which ensures that:

$$| d_j^{(n)} - d_j^{(m)} | < \epsilon$$

Since for any $\epsilon>0$ we can find an N such that for all $m,n \geq N$ the difference $| d_j^{(n)} - d_j^{(m)} |$ is arbitrarily small, the sequence is *Cauchy*. Following a fundamental theorem in Complete Metric Spaces, every Cauchy sequence converges (see Rudin (1976) for a detailed proof of the theorem). This entails convergence of the AQER process ■

APPENDIX C – THE IMPACT OF AQER'S MAIN COMPONENTS

In this appendix, we report on an ablation study designed to evaluate the impact on the performance of AQER's two main components: the multidimensional voting and the iterative reweighting procedures. To this end, we compared the performance of a full implementation of the AQER framework against the performance of a simplified implementation in which the reweighting procedure was "turned off", leaving the framework with only the multidimensional voting mechanism. As observed in Table C1, the multidimensional voting method by itself obtains very good performance even without using the iterative reweighting procedure. Nevertheless, activating the iterative reweighting procedure provides an additional 1%-2% performance improvement, which was statistically significant in all cases ($p < 0.01$). It is important to note that this improvement by the iterative reweighting procedure is not trivial given the already very good results obtained by AQER's multidimensional voting component, and the minor possibilities for improvement when the Pearson correlation is already close to 1. Furthermore, as shown in Appendix F, in some more challenging conditions the iterative reweighting component of AQER provides additional robustness and contributes to significant performance improvement. Overall, the results of this ablation

study indicate that the multidimensional voting concept is the more impactful component of our approach (it is also a pre-requisite for applying the iterative reweighting component), yet when accuracy is paramount, or when conditions are more challenging (as shown in Appendix F), it is recommended to apply both of AQER's components to obtain the best possible performance.

Table C1. Ablation Study, The Impact of Iterative Reweighting

A	*B*	*C*	*D*
Settings	AQER Using Multidim. Voting (Iterative Reweighting Turned Off)	Full AQER Using Multidim. Voting and Iterative Reweighting	%Improvement by using Iterative Reweighting
Science & Sports	0.939	0.950	1.2%***
History & Movies	0.898	0.915	1.9%***
Computer Science	0.945	0.963	1.9%***

Note: Columns B and C in this table present the Pearson correlation coefficient between each worker's average response grade as determined by human expert evaluators and the worker's grade calculated using our AQER framework with two different implementations (either with or without iterative reweighting). Column D reports the percent of improvement when using multidimensional voting in AQER. The reported results for the semi-synthetic Computer Science dataset represent the average of 100 simulation repetitions. *** indicates that the mean difference between the AQER framework implemented with and without multidimensional voting is statistically significant with a p-value<0.01 (calculated using BCA bootstrap).

APPENDIX D – ADDITIONAL PERFORMANCE MEASURE

In the main body of the paper, we reported results for the worker evaluation task using the Pearson correlation coefficient which is commonly used in related ASAG research. For robustness, in this appendix we report the results obtained for the same datasets, but with an alternative performance measure: the cruder[4] Spearman rank correlation coefficient. Specifically, Table D1 reports the performance of our method and the baselines using this measure. The results, given this additional metric, show that AQER is still the method of choice and that it provides robust performance for the worker evaluation task: it is either substantially better or at least comparable to the best alternative.

[4] Spearman rank correlation coefficient is generally cruder than Pearson correlation. Spearman rank correlation only measures the relative rankings between the workers' rank and their true rank, and it does not consider the magnitude of the differences.

Table D1. Worker Evaluation - AQER and Baseline Performance (Spearman Rank Correlation coefficient)

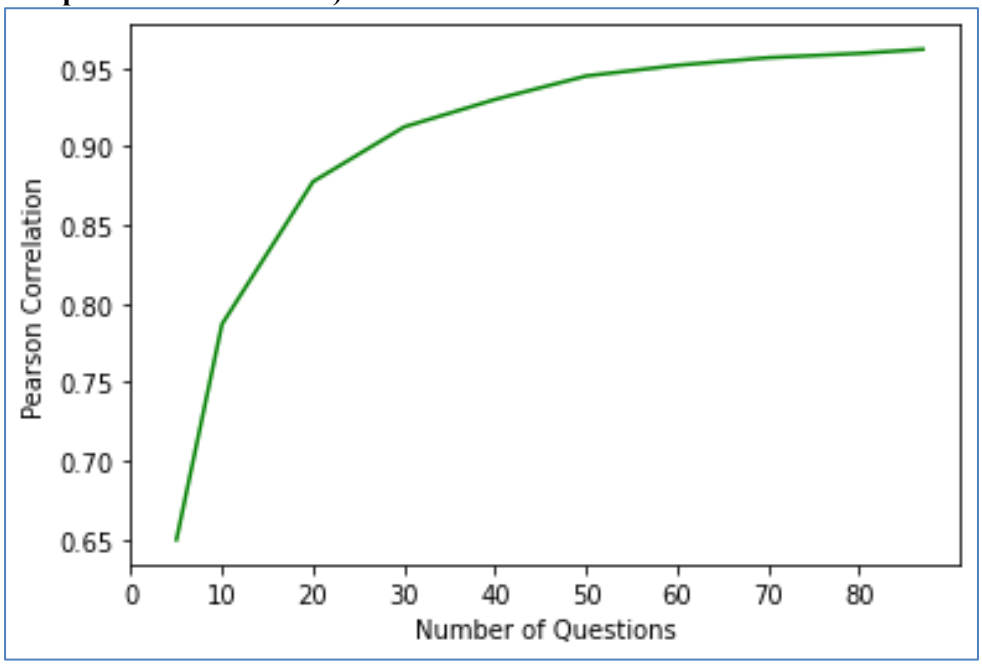
Dataset	**AQER (Ours)**	Roy et al., 2016	Li & Fukumoto, 2019	Chai et al., 2022	Bleu1	Bleu4	Bert Score	F1	LLM as a judge
Science & Sports	0.945	0.865	0.945	-0.029	0.683	0.695	0.629	0.747	0.890
History & Movies	0.906	0.779	0.775	-0.087	0.612	0.528	0.441	0.594	0.808
Computer Science	0.883	0.847	0.884	0.055	0.787	0.808	0.806	0.842	0.890

This table presents the Spearman rank correlation coefficients between each worker's average response grade as determined by human expert evaluators and the worker's grade calculated using our AQER framework and various baseline approaches. The reported results for the semi-synthetic Computer Science dataset represent the average of 25 simulation repetitions.

APPENDIX E – THE IMPACT OF THE NUMBER OF QUESTIONS

We evaluated how the number of questions affects the performance of the AQER framework. The capacity to identify high-quality workers with a small number of questions may be useful towards reducing the cost of initial screening of workers. In this analysis, we drew upon the semi-synthetic-based simulation approach (computer science dataset). Recall that the original dataset has 87 questions. Based on the original dataset, we simulated additional datasets having [80, 70, 60, 50 ,40 ,30, 20, 10, 5] questions by randomly omitting questions from the original dataset. The simulation procedure was repeated 50 times, and in each iteration, different questions were removed. Figure E.1 reports the mean Pearson correlation value between the AQER-generated scores and the human-generated scores. As is evident in this figure, AQER performed well even when the number of questions was much lower than in the original dataset. For example, when supplied with 30 questions, AQER obtained a mean Pearson correlation above 0.9; and when the number of questions was reduced to only 10, AQER obtained a mean Pearson correlation of 0.787 (p-value was lower than 0.05, in 49 out of 50 simulation repetitions).

Figure E1. AQER Performance as a Function of the Number of Questions (Evaluation Based on Computer Science Dataset)



Note. This figure presents the mean Pearson correlation (across 50 simulation runs) between each worker grade as determined by human evaluators (where each worker's grade is the average score across that worker's individual responses) and the worker grade calculated by the AQER framework. The x-axis represents the number of questions drawn at random from the semi synthetic-Computer Science dataset, on the basis of which AQER conducted its scoring.

APPENDIX F – NUMERICAL SIMULATION

In AQER's analytical motivation (section 4.1.1), we derived the boundaries of the error of the SEA (versus the corresponding correct response vector) for cases in which standard PAC-learning-based assumptions are met, and in Online Appendix B3 we proved that under these assumptions, AQER's iterative process converges. In this appendix, we present the simulation procedure we developed for testing the impact of relaxing the model's assumptions and additional extreme conditions.

The simulation scenarios described below, unless stated otherwise, involve 30 workers, who each answer the same $n = 20$ questions. For each question Q_j, we simulate a correct response vector, $text_{true,j}$, of size 512. Specifically, the values of the vector elements of the correct response vector, $text_{true,jk}$ (where k=1,…,512), are randomly generated with normal distribution with a mean of 0 and standard deviation of

1. In addition, for each worker w_i, we create 20 responses; each response corresponds to one of the 20 questions. Worker w_i's response to question Q_j is represented by vector $text_\text{ij}$ with 512 elements, whose kth element is denoted by $text_\text{ijk}$. $text_\text{ijk}$ is also randomly generated using a normal distribution with a default mean that is equal to $text_{true,jk}$, and a simulation-defined standard deviation and bias (to be detailed in each specific simulation scenario). This approach enables us to generate workers with known individual "true" s_i (calculated by $s_i = \frac{1}{n}\sum_j cosine(text_{true,j}, text_\text{ij})$.

Each simulation scenario was repeated 30 times to derive average outcomes.

Finally, in the simulations below we typically consider two types of workers: "high-quality workers" and "low-quality workers" that may differ by their response standard deviation and bias. In the different scenarios, we manipulate the standard deviation and bias values or the ratio and absolute numbers of each type of worker.

In the following sections we will evaluate the impact of gradually relaxing the following assumptions:

(1) Workers are independent: $Cov(text_\text{ijk}, text_\text{ljk}) = 0$, for all $i \neq l, j, k$

(2) *M* (the number of workers) is large.

In addition we evaluate the impact of other factors:

(3) The variance in the responses' representations;

(4) Bias in workers' responses.

F1. Violating the Assumption that Workers' Responses are I.I.D

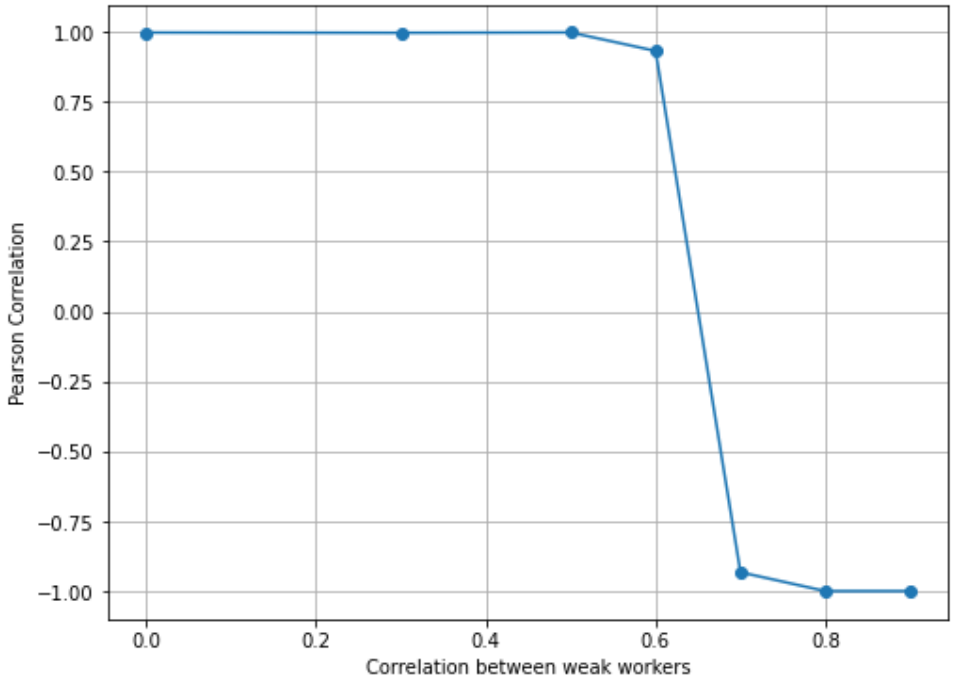
In this section, we test the impact of relaxing the assumption that the workers produce independent responses. Specifically, we seek to test the performance of AQER in challenging conditions in which "low-quality workers", who produce inaccurate responses, also tend to produce the same inaccurate responses.

This scenario may occur in real-world settings in which, for example, multiple workers provide a similar incorrect response due to shared background; workers are being adversarial and intentionally coordinate to produce a similar, incorrect, response; or multiple workers resort to providing some default incorrect answer such as “I don’t know”.

This simulation considers two types of workers: either high-quality workers (with zero bias and a standard deviation of 0.5 for each response element $text_{\mathrm{ijk}}$), or low-quality workers (with zero bias but with a standard deviation of 3 for each element $text_{\mathrm{ijk}}$). The simulation begins when there is no correlation among the low-quality workers, and gradually increases the Pearson correlation (from 0 to 1) between the low-quality workers to simulate scenarios where workers are not independent. We note that the simulation procedure produces low-quality workers with both higher variance (std of 3) and increasingly growing levels of correlation between them.

Figure F1 presents these simulation results. The x-axis presents the level of Pearson correlation among low-quality workers, while the y-axis shows the framework's performance (Pearson correlation between workers’ grades according to AQER, and the workers’ true grades). In the scenario presented, half the simulated workers are high-quality workers and half the workers are low-quality workers (whose internal correlation is being gradually increased). In this figure, we observe that the AQER model achieves high performance as long as the correlation between the responses of low-quality workers is lower than 0.7. However, when the correlation reaches 0.70, the performance drops significantly.

Figure F1. The performance of AQER when the correlation among low-quality workers is gradually increased, with half of the workers being of low quality.



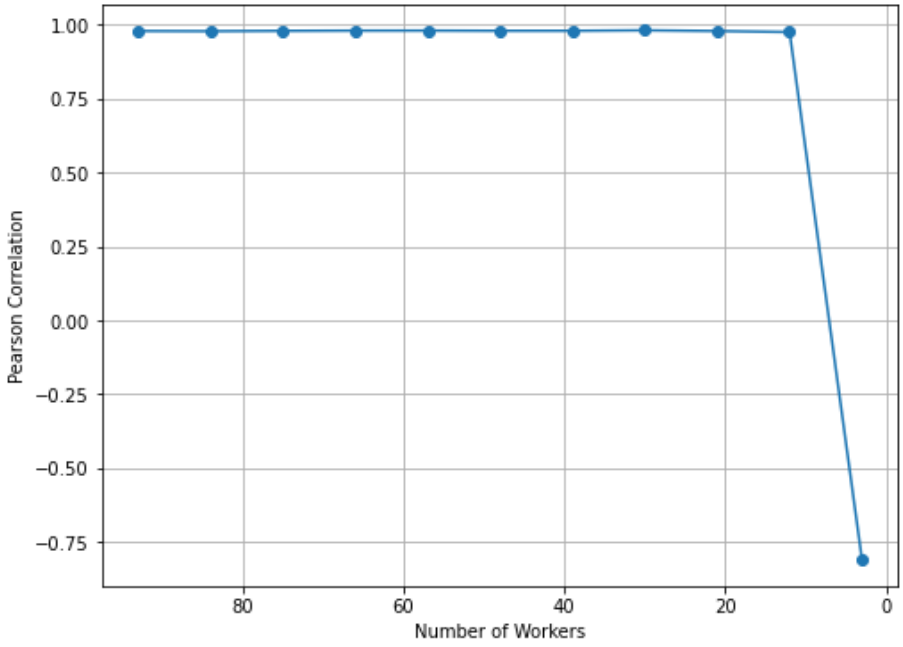
F2. Violating the Assumption that the Number of Workers Is Large

To evaluate the impact of this assumption, we begin the simulation with 90 workers; a third of these workers are simulated to be high-quality workers with a standard deviation of 0.5, a third are and the workers are simulated to be medium-quality workers with a standard deviation of 1, and a third of the workers are simulated as low-quality workers with a standard deviation of 2. We then gradually reduced the number of workers until there are only 3 workers left.

Figure F2 shows the performance of AQER when 50% of the workers are high quality and the rest are low quality, and as the number of workers gradually decreases.[5] As observed, the model produces excellent results until the number of workers reaches 3.

[5] This simulation reduces the same ratio of low-quality workers and high-quality workers in each iteration.

Figure F2: The performance of AQER when decreasing the number of workers, with 50% of the workers being of low quality.

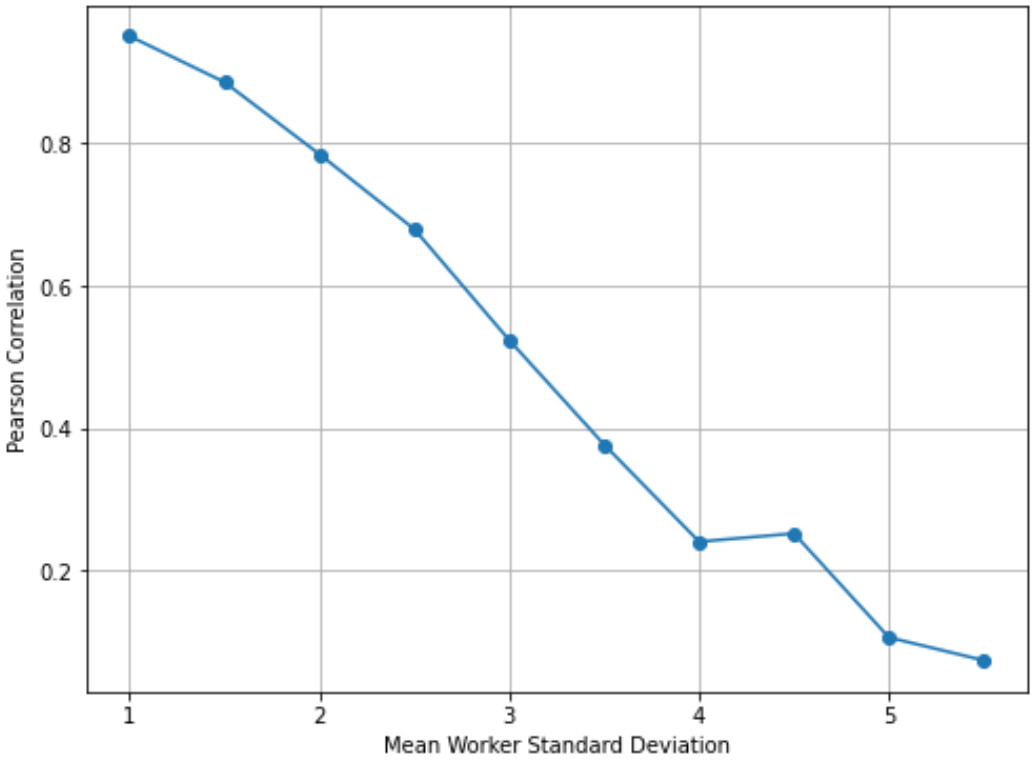


This analysis indicates that the model is robust to reductions in the number of workers until a critical threshold is reached.

F3. Noisy Responses - High Embedding Variance

To test the impact of noisy responses, resulting in high embedding variance, we begin our simulation with three types of workers: high-quality workers (with zero bias and a standard deviation of 0.5 for each element $text_{\mathrm{ijk}}$), medium-quality workers (with zero bias and a standard deviation of 1 for each element $text_{\mathrm{ijk}}$), and low-quality workers (with zero bias and a standard deviation of 1.5 for each element $text_{\mathrm{ijk}}$). We then gradually increase these standard deviation values by increments of 0.5 until the standard deviation increase reaches 5. Figure F3 illustrates the performance of AQER as the mean standard deviation across all workers' $text_{\mathrm{ijk}}$ elements increases.

Figure F3: The performance of AQER when increasing the standard deviation of worker responses



The results show that the model starts with a high Pearson correlation (between AQER and true grades) that is close to 1.0 when workers' response standard deviation is low. As the standard deviation increases, the performance of the model declines.

F4. Noisy Responses - Systematic Bias

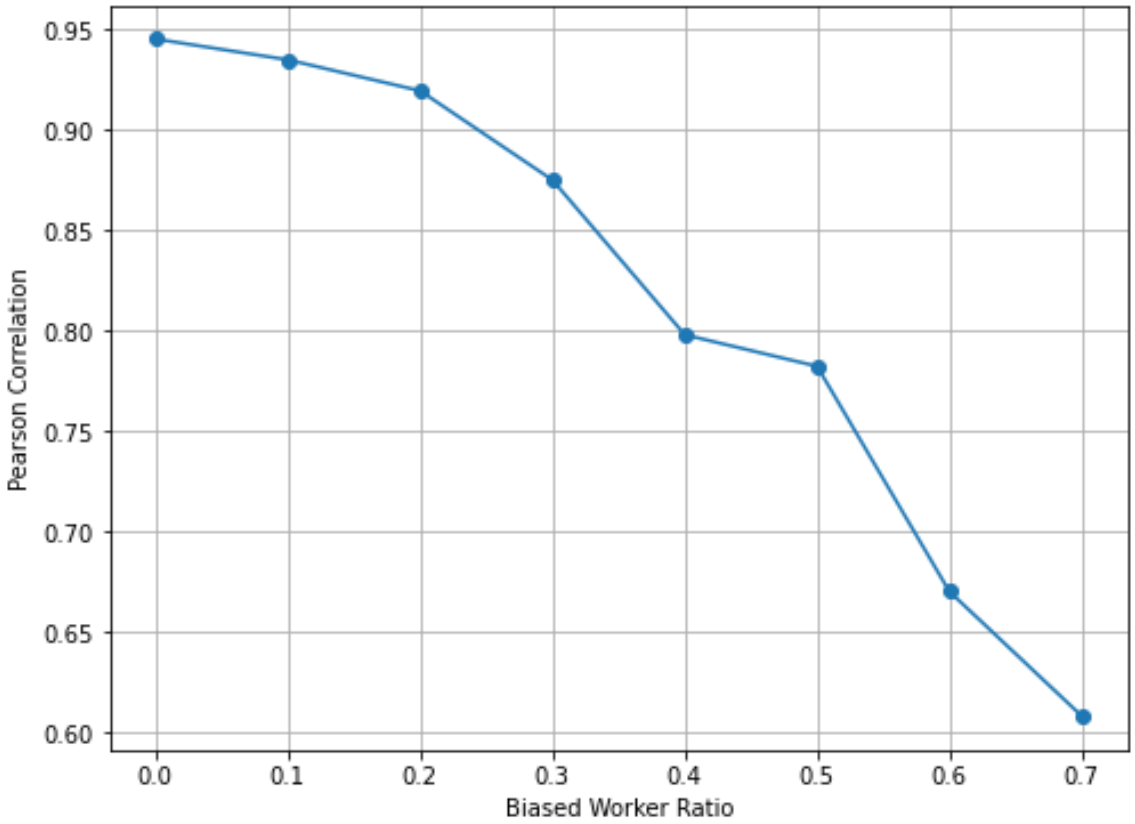
We sought to test how AQER handles systematic bias, i.e., biases with a shared direction. We began by simulating half of the workers as "high-quality workers" with zero bias and a standard deviation of 0.5 and half of the workers as "medium-quality workers with zero bias and a standard deviation of 1 for each response element $text_{\mathrm{ijk}}$. We then gradually introduced "low-quality workers". Specifically, we repeatedly replaced one "high-quality worker" with one "biased" worker whose response vector elements, $text_{\mathrm{ijk}}$, have a systematic bias of 0.5 (the standard deviation of 0.5 is unchanged) and one "medium-quality worker" with

one “biased” worker whose response vector elements, $text_{ijk}$, have a systematic bias of 0.5 (the standard deviation of 1 is unchanged).

For each percentage of systematically-biased workers, we calculated AQER’s performance using the Pearson correlation between the grades produced by AQER and the workers' true grades. The simulation results are presented in

Figure F4. In this figure, the x-axis represents the percentage of biased workers in the entire set, while the y-axis shows the framework's performance (where performance is evaluated according to the Pearson correlation between the workers’ true grades and the grades calculated by the framework).

Figure F4. The performance of the AQER model when increasing the percentage of biased workers.



As shown in Figure F4, AQER maintains high performance (high Pearson correlation) initially but begins to degrade noticeably when the proportion of biased workers exceeds approximately 30%. It is important to

note, however, that a scenario that involves systematic biases in embedding vectors is likely to be very uncommon in reality, unless the workers are adversarial, conducting a systematic attack. In case of non-systematic biases, the biases would mathematically cancel each other out.

APPENDIX G – COMPARISON WITH BAG-OF-WORDS REPRESENTATION AND MAJORITY VOTING

In this appendix, we evaluate the impact of the textual representation and the corresponding multidimensional voting scheme (either majority voting or average voting). As detailed in section 4.1, AQER may be implemented with either average voting or majority voting.

Table G1 compares the performance of a basic AQER implementation, with only the voting component (the iterative EM-based reweighting component is turned off), either using a binary BOW representation[6] and corresponding majority voting or using RoBERTa-based embedding (our standard implementation) with average voting. The table shows that using advanced textual representation with the corresponding average voting outperforms using BOW representation with the corresponding majority voting.

Table G1. Worker Evaluation – AQER Without Iterative Re-Weighting, Using Different Textual Representation and Voting Options

Dataset	BOW-Based Representation with Majority Voting	Embedding-Based Representation with Average Voting
Science & Sports	0.859	**0.939*** **
History & Movies	0.609	**0.898*** **
Computer Science	0.879	**0.945*** **

This table presents the Pearson correlation coefficient between each worker's average response grade as determined by human expert evaluators and the worker's grade calculated using our AQER framework with the iterative re-weighting procedure turned off, using two different implementations: either using a BOW-based representation with majority voting or using an embedding-based representation with average voting. The reported results for the semi-synthetic Computer Science dataset represent the average of 100 simulation repetitions. *** indicates that the mean difference between the two alternatives is statistically significant with a p-value<0.01 (calculated using BCA bootstrap).

Table G2 shows a similar comparison while using the full AQER implementation, i.e., with the

[6] Each word in the text is lemmatized using NLTK package in python. If a lemmatized word appears within the text then the binary vector element representation for this word will indicate '1', otherwise '0'.

iterative EM-based reweighting component turned on. In this case as well, the table shows that using advanced (RoBERTa-based) textual representation with corresponding (weighted) average voting outperforms using BOW representation with corresponding (weighted) majority voting. Overall, the results demonstrated in both tables show that it is clearly advantageous to implement AQER with more advanced textual representations, such as transformer-based embedding, together with the corresponding average voting scheme.

Table G2. Worker Evaluation – Full AQER, With Iterative Re-Weighting, Using Different Textual Representation and Voting Options

Dataset	BOW-Based Representation with (Weighted) Majority Voting	Embedding-Based Representation with (Weighted) Average Voting
Science & Sports	0.901	**0.950*** **
History & Movies	0.625	**0.915*** **
Computer Science	0.930	**0.963*** **

This table presents the Pearson correlation coefficient between each worker's average response grade as determined by human expert evaluators and the worker's grade calculated using our AQER framework with the iterative re-weighting procedure turned off, using two different implementations: either using a BOW-based representation with majority voting or using an embedding-based representation with average voting. The reported results for the semi-synthetic Computer Science dataset represent the average of 100 simulation repetitions. *** indicates that the mean difference between the two alternatives is statistically significant with a p-value<0.01 (calculated using BCA bootstrap).

APPENDIX H – BASELINES IMPLEMENTATION

As discussed in section 5.4, we compare AQER's performance to the performance of multiple baseline approaches. In this appendix, we provide details on the implementation of these baseline approaches.

The Bleu1 and Bleu4 measures were used as baselines for comparison following studies that compiled Q&A datasets (e.g., Nguyen et al., 2016; Bajaj et al., 2018; Kočiský et al., 2018). These studies did not develop methods to assess worker quality or score LLM responses. Instead, they employed machine translation measures, in an ad-hoc manner, for internal consistency verification (see the discussion in the "Related Work" section). We followed these works and constructed baselines using the same machine translation measures (Papineni et al., 2002). Specifically, similarly to these studies, for the worker evaluation task, we calculated the Bleu score for each focal answer by considering all other answers as "correct" reference answers. Similarly, for the language model response evaluation task, we calculated the

Bleu score for each LLM response while considering all of the workers' responses as reference answers. To calculate the Bleu scores (either Bleu1 or Bleu4) we used word tokenization and Bleu score calculations using Python's popular NLTK package.

BERTScore (Zhang et al., 2019) is a more recent algorithm that allows, similar to the Bleu1 and Bleu4 scores, to evaluate the extent a reference text matches the responses or translation provided in multiple candidate answers. Unlike BLEU metrics, which rely on exact word matching, BERTScore leverages contextualized token embeddings to measure semantic similarity between texts. In contrast to our method—which compares entire sentence embeddings using cosine similarity—BERTScore constructs a cosine similarity matrix by comparing each token from the candidate text with every token in the reference text, ultimately deriving a similarity score from the best-matching token pairs.

We used BERTScore as an additional baseline. Specifically, we used the bert-score python package with its default configuration, which relies on RoBERTa embedding and returns the best-matching F1 score. For the worker evaluation task, we used the F1 measure that is internally reported by the BERTScore algorithm for each focal answer by considering all other answers as "correct" reference answers. Similarly, for the language model response evaluation task, we calculated the BERTScore for each LLM response while considering all of the workers' responses as reference answers.

Another baseline that we used is based on the paper by Roy et al. (2016). Specifically, to calculate a score for each individual response, we implemented and optimized the method proposed by Roy et al. (2016). We note that their paper does not provide exact implementation procedures in an algorithm block, and does not include a sharable implementation code[7], and hence there is insufficient information regarding the maximal length of the sequences that they used.[8] Therefore, to produce a meaningful baseline, we optimized Roy et al.'s (2016) approach by selecting the most promising implementation after testing multiple sequence lengths, and we report the performance of this implementation in this paper. Specifically, we randomly selected 30 questions from the dataset compiled by Mohler et al. (2011) and tested

[7] We have contacted the authors of (Roy et al., 2016) asking for the code. However, they did not have the code available.
[8] We note that the method by Roy et al. (2016) is extremely computationally intensive (or practically infeasible) for long sequences; thus, determining the sequence length used in practice is vital for any implementation of this approach.

implementations with variable maximal sequence lengths.[9] We then implemented the best-performing setup as the baseline approach throughout our evaluations. Additionally, to implement the stemming and stop-word filtering procedures, we used the popular NLTK package in Python. We note that optimizing the baseline's sequence length over a part of the dataset (Mohler et al., 2011) that was used in our evaluations may provide an advantage to Roy et al.'s (2016) baseline approach. Thus, the improvements over the baseline for this dataset (if obtained) may be regarded as conservative estimates. Additionally, the method by Roy et al. (2016), was not designed, and does not include a mechanism to evaluate external responses such as LLM responses. Therefore, to evaluate the LLM responses, we selected the highest-scored response by Roy et al.'s method as a reference answer and scored the LLM response using the Bleu1 measure with this reference answer.

To implement LLM-as-a-judge, we presented OpenAI's ChatGPT-4o-mini model with a pair consisting of a question and an answer in each session. We then asked the model, acting as a "judge," to evaluate the quality of the answer and provide a numeric score between 0 and 100. As a safeguard against rare instances in which the model may fail to produce a numeric score, our code included a verification mechanism to ensure that the response was numeric. If the response is not numeric, the model is prompted again. If, after two attempts, it still fails to generate a numeric score, the code assigns a default value of 50 (the midpoint between 0 and 100). Before deploying the "LLM-as-a-judge" approach with ChatGPT-4o-mini, we manually reviewed its outputs and our evaluation logs across two simulation runs using all three datasets. We did not encounter any instances where it failed to generate numeric scores. Additionally, we used data from these evaluations to compare ChatGPT-4o-mini's ability to evaluate workers and LLMs against that of the more expensive ChatGPT-4o. The observed differences in performance between the two models in evaluating workers and LLMs were negligible, leading us to conduct large-scale testing with the more cost-effective ChatGPT-4o-mini.

Importantly, as noted in the main body of the paper, it is important to acknowledge that ChatGPT-4o-mini (and ChatGPT-4o) may have an inherent advantage when serving as a baseline for the CS dataset.

[9] Specifically, we tested maximal sequence lengths of 2, 3, 5, 10, 15 (stemmed) words. We found that the best results were produced with a maximal sequence length of 2, and that in general, the shorter the sequence length used by the baseline, the better.

Since the dataset's questions, answers, and responses are publicly available online, they may have been included in the training data for these models.

Additionally, we followed (Liang et al., 2023) and used the F1 metric for text similarity. Different from BERTScore, in this case, the F1 score was directly based on word similarity in the text between a focal answer and reference answers for evaluating the correctness of a response. Specifically, we use Python's NLTK package to calculate the F1 score between a focal answer and each reference answer and return the maximum score.

To implement and adapt the method proposed by Li and Fukumoto (2019) we primarily relied on the publicly shared code for their RASA approach and made only minimal modifications required to enable the code to run on a recent Tensorflow version and to employ USE embedding from Tensorflow hub. Additionally, recall that the method proposed by Li and Fukumoto (2019) did not aim to produce worker evaluations or to score LLMs. Therefore, to address the worker evaluation task we extracted reliability scores which are internally calculated by Li and Fukumoto's method. To address the LLM evaluation task we generated an embedding vector for the "best" response (for each question) as selected by Li and Fukumoto's method. We then scored each LLM response according to its cosine similarity to an embedding vector of the selected response by Li and Fukumoto's method.

Finally, the paper by Chai et al., (2022), which we also used as a baseline, does not have a publicly available code.[10] We therefore implemented their approach based on the description in their paper. To implement their gradient descent approach, we used a numerical differentiation, a step size of 0.001, and an early stopping condition.[11] Since Chai et al., (2022) did not aim to evaluate workers we extracted internal response error scores for each question and worker combination and graded each worker according to the negation of each worker's average response error. To address the LLM evaluation task, again, we generated an embedding vector for the selected response to each question and scored each LLM response according to its cosine similarity to the relevant embedding vector.

[10] We have contacted the authors asking for the code however we did not get a reply.
[11] We also tested smaller step sizes and saw only negligible or no improvement, with substantial increase in run time.

APPENDIX I – BEST RESPONSE SELECTION

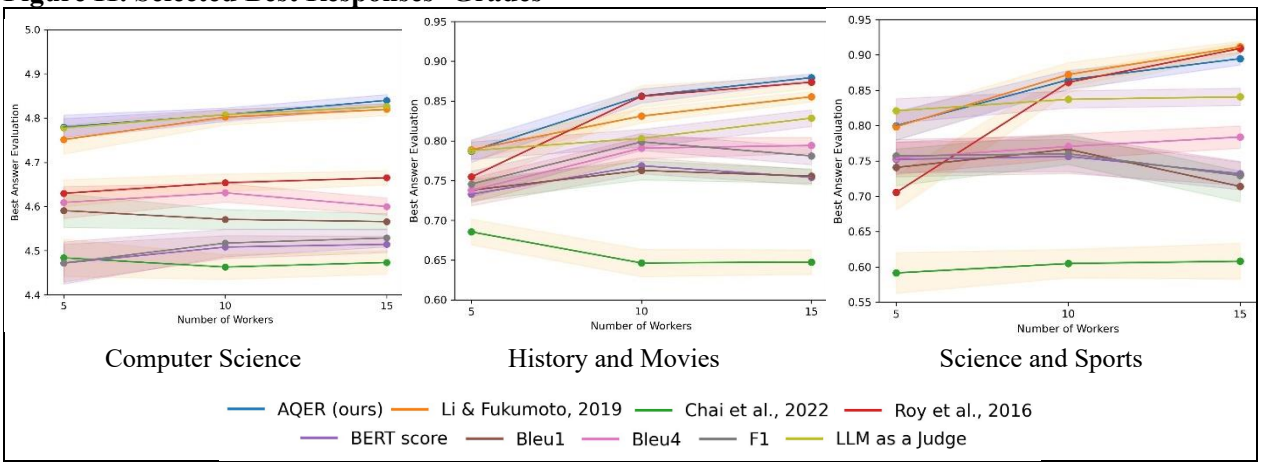
The original goal of some of the baseline approaches (e.g., Li and Fukumoto, 2019; Chai et al., 2022) was to select the best response for each question rather than to evaluate LLMs. Although best response selection is not one of AQER's main goals, in this Appendix we evaluate how AQER performs at the task of best response selection.

In particular, to address the goal of selecting the best worker-generated response to the question Q_j, we make another use of SEA_j (previously calculated using Algorithm 1 in the main body of the paper). As SEA_j cannot be directly transformed into a textual response, AQER can employ a simple-to-implement procedure to select the most accurate worker-generated response. Specifically, for each $text_{i,j}$ (the vector representing a response $R_{\mathrm{i,j}}$ for question Q_j) AQER can calculate the similarity between$text_{i,j}$ and SEA_j. AQER can then choose the response $R_{\mathrm{i,j}}$ that corresponds to the highest similarity. In this work, we consider the popular cosine similarity measure. However, other similarity measures may be used as well (e.g., inverse Euclidian distance).

Similar to LLM response evaluation, we evaluate best response selection beginning with a small number of randomly selected workers (M=5), and then gradually increasing the number of workers recruited to answer each question (considering also M = 10, 15). For each value of M, we repeated the evaluation 25 times; we then applied AQER and the baselines to select the best responses, and we documented the real grade[12] of each selected response. The mean real grades for the best responses selected by implementing our AQER approach and the baselines are presented in Figure I1, together with 95% confidence intervals. As observed, the AQER method is robust: it consistently achieves either superior results compared to the baseline approaches or is otherwise equivalent to the best alternatives. These results demonstrate the utility of AQER in extracting particularly high-quality responses out of all the responses provided by the crowd.

[12] The real grade is the human evaluator-based grade that is strictly kept hidden from AQER and the baselines. Note that each response was originally graded by two evaluators as discussed in section 5.2 and Online Appendix A.

Figure I1. Selected Best Responses' Grades



Note: The figure shows the mean (real) grade and 95% confidence intervals (lightly shaded colors) for the best response selected by implementing AQER and the baselines. Additionally, the figure presents the average grade of all of the responses provided by the selected workers. Results are based on 5, 10, 15 randomly selected workers from each dataset. For each number of workers selected, the results are based on 25 repetitions of random sampling of workers. Grades for the "Science and Sports" and "History and Movies" dataset are between 0% and 100%. Grades for the Computer Science dataset (Mohler et al., 2011) are between 0 and 5.

APPENDIX J – COMPARISON TABLE

The Related Work section discusses several papers that address related problems. Table J1 provides a comparison table for the main differences and similarities between our work and the most relevant related works.

Table J1: Comparison with Related Works

	Approach				Goals			Evaluation		
Paper	Modular Framework	Provides Required Assumptions	Uses an Iterative procedure	Unsupervised Approach	Goal: Assessing Language Models	Goal: Worker Evaluation	Goal: Extract Correct or Aggregated Response	Empirical Evaluation using Real-World Datasets	Uses purposely built numerical simulation to assess operating conditions	Analytical Evaluation
Li and Fukumoto, 2019	✗	✗	✓	✓	✗	✗[2]	✓	✓	✗	✗
Li, 2020	✗	✗	✓	✓	✗	✗[2]	✓	✓	✗	✗
Chai et al., 2022	✗	✗	✓	✗	✗	✗[2]	✓	✓[6]	✗	✗
Braylan and Lease, 2020	✓[1]	✗	✓	✗	✗	✓	✓	✓[6]	✗	✗
LLM judge (e.g., Zheng et al., 2023)	✗	✗	n/a[7]	✓[7]	✓[7]	✗	✗	✓	✗	✗
Nguyen et al., 2016	✗	✗	✗	✓	✓	✗	✗	✓	✗	✗
Kočiský et al., 2018	✗	✗	✗	✓	✓	✗	✗	✓	✗	✗
Roy et al., 2016	✗	✗	✗	✓	✗	✓[3]	✗[4]	✓	✗	✗
AQER (our work)	✓	✓	✓	✓	✓	✓	✗[5]	✓	✓	✓

Notes:

1. The approach modularity is restricted to supporting several metrics.
2. This is not a goal of the approach, but worker evaluation can be obtained from internal measures used by the approach.
3. The paper scores each response separately so workers' evaluation can be conducted using the average score.
4. The paper does not aim to select the best response however this information can be obtained.
5. Although it is not a primary goal of our work - our AQER framework can extract the best response out of the set of existing responses. This is demonstrated in Appendix I.
6. The paper focuses on other forms of text (e.g., translation) and does not evaluate free-text responses to questions.
7. LLM as a judge is not supervised or fine-tuned for the evaluation of LLMs or workers' responses. It also does not use a dedicated iterative procedure for improving its LLM evaluations.

APPENDIX K – INCREASING A HIGH-QUALITY WORKER'S NUMBER OF INCORRECT RESPONSES

In this appendix, we report on an analysis that builds on simulation settings described in Appendix F and examines the impact on the performance of AQER when a 'high-quality' worker provides incorrect answers to an increasing number of questions.

In this simulation scenario, 50% of the workers are 'high-quality' (standard deviation of 0.5) and 50% are 'low-quality'(standard deviation of 3). We gradually increased the number of incorrect responses from one of the 'high-quality' workers up to a total of ten incorrect responses.[13] We observed that the model produces excellent results with *less than 1% degradation*.

APPENDIX L – ADDITIONAL PROMPTING STRATEGY

As discussed in the main body of the paper, AQER is not only useful for comparing LLMs' accuracies but can also be used to evaluate the accuracy of LLM responses given different prompting strategies. In this appendix, we provide a simple demonstration of AQER's usefulness, compared to the baseline approaches when using an alternative prompting strategy with one of the LLMs used in the main body of the paper (Lamini LLM).

The original prompting format used in the main body of the paper was: "*With regards to <topic>, please answer the following question in one sentence: <the question's text>. Your output format should be: 'The answer is: {{generated answer}}*".[14]

For an alternative (simple) prompting strategy, we used the following prompt: "*With regards to <topic> please answer the following question as concise as possible: <the question's text>. Your output format is 'answer: {{short answer}}'* ".

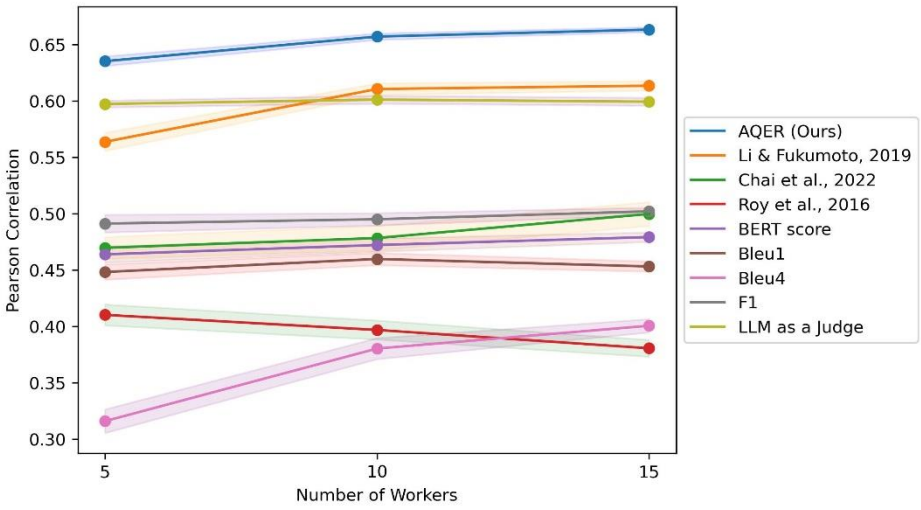
Performance evaluation was conducted in a manner similar to that reported in Section 6.2, using 5, 10,

[13] To create these incorrect responses, we increased the worker's standard deviation by 10, from 0.5 to 10.5, for these specific questions, resulting in a cosine similarity close to zero with the correct responses.

[14] For example: With regards to the Ironman race, please answer the following question in one sentence: Why do different Ironman races have different time limits? Your output format should be: 'The answer is: {{generated answer}}"

and 15 workers. As observed in Figure L1, AQER continues to achieve superior performance over the baseline models even when applying this alternative prompting strategy.

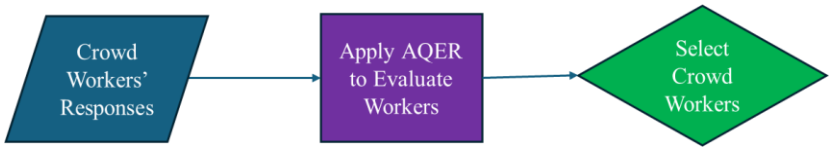
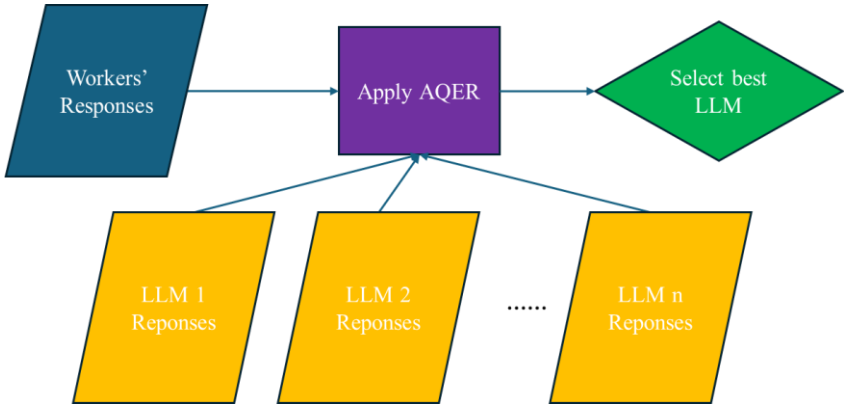
Figure L1. Evaluation of the Lamini LLM when Responses Were Generated Using a Different Prompt

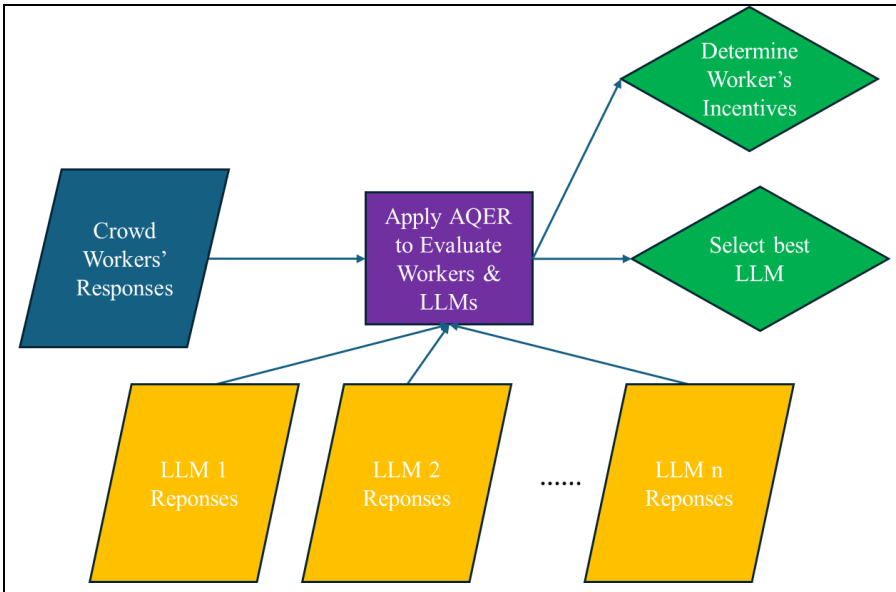


APPENDIX M – GRAPHICAL ILLUSTRATION OF AQER USAGE SCENARIOS

In this appendix, we provide a visual illustration of a few of AQER's use cases. Each Illustration is accompanied by explanations in Table M1

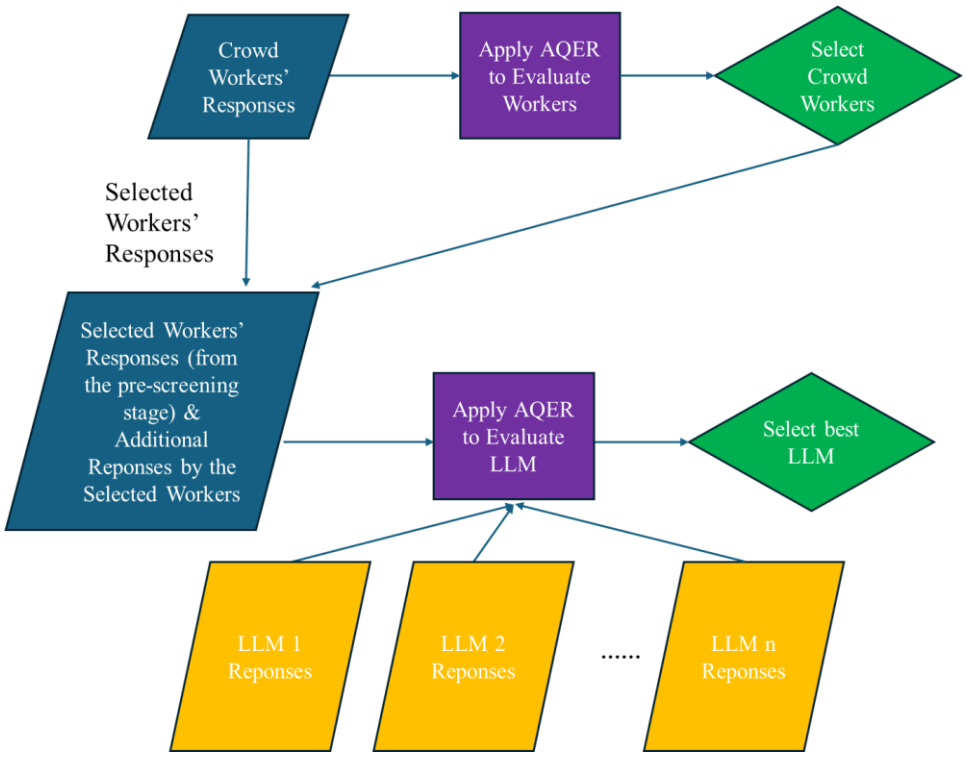
Table M1: A Graphical Illustration of Several AQER Usage Scenarios

Scenario 1: Using AQER for Pre-Screening of Potential Crowd Workers In this scenario, AQER is applied to the responses of a group of crowd workers, who respond to a set of questions in a given domain. The top performing (most accurate) workers are selected and may be called upon in the future to answer additional questions in this domain.   
Scenario 2: Using AQER for LLM Model Selection (given pre-selected workers) This scenario assumes that the workers have already been selected and provided responses to a set of questions. The workers may be crowd workers, who were found to be reliable in the past, expert workers within the organization, or workers previously selected by AQER. AQER is then applied to evaluate the accuracy of different LLMs' responses. Response evaluation could be conducted for different LLM models, different development versions, or various fine-tuning procedures.   
Scenario 3: Using AQER for LLM Model Selection and Simultaneously Monitor Workers' Response Quality to Incentivize Them This scenario also assumes that the workers have already been selected and provided responses to a set of questions. However, in the current scenario, AQER is applied for two purposes. The first is to evaluate the LLMs. The second is to provide an ongoing evaluation of the workers. This evaluation could then be used to verify the quality of the workers and to determine incentives for high-performing workers.



Scenario 4: Using AQER for Pre-Screening of Potential Crowd Workers and Reusing the Responses for LLM Evaluation

In this scenario, crowd workers are first pre-screened using AQER (similar to scenario 1). However, the responses of the selected crowd workers from the pre-screening phase are reused. These responses, together with additional responses from the selected workers are subsequently used by a second instantiation of AQER to evaluate the accuracy of LLM responses.



APPENDIX N – ROBUSTNESS - ADDITIONAL SEMI-SYNTHETIC SIMULATIONS

The Computer Science (CS) dataset used in this paper involves a simulated assignment to individual workers of the responses reported in a real-world Q&A dataset. This simulation procedure was reported in section 5.1 of the paper. However, we acknowledge that in practice there are many possible ways to simulate the response assignment. For robustness, we report the performance of two variations of the original simulations in which, on average, 20% or 30% of the workers' responses are randomly assigned. Table N1 reports the results for the original CS simulation as well as the two new variations. As reported in Table N1, increasing the randomness of the response assignment generally results in a small degradation in performance for AQER and the baseline approaches. However, as observed, even when increasing the randomness AQER continues to be the method of choice, displaying robust performance, regularly significantly surpassing benchmark approaches or, at minimum, matching the best-performing baseline (LLM-as-a-judge). These results match our findings using the CS dataset in the main body of the paper.

Table N1. Worker Evaluation - AQER Performance versus Baseline Approaches, Different Semi-Synthetic Simulation Settings

Dataset	AQER (Ours)	Roy et al., 2016	Li & Fukumoto, 2019	Chai et al., 2022	Bleu1	Bleu4	BertScore	LLM as a Judge
CS Dataset – Original Simulation	**0.964**	0.924	0.927	0.055	0.870	0.864	0.895	0.960
CS Dataset – 20% Random Assignment	**0.948**	0.895	0.909	0.030	0.811	0.794	0.835	0.946
CS Dataset – 30% Random Assignment	**0.936**	0.868	0.892	0.076	0.776	0.750	0.789	0.930

Note. This table presents the Pearson correlation coefficients between each worker's average response grade as determined by human expert evaluators and the worker's grade as calculated using our AQER framework and various baseline approaches. The reported results represent the average of 25 simulation repetitions.

APPENDIX O – EXAMPLES OF QUESTIONS AND RESPONSES

In this appendix, we present examples of responses to two questions from the "Movies and History" dataset, along with evaluations of these responses by human expert evaluators and by AQER. We focus on question difficulty and we intentionally select questions that represent contrasting cases: one with the most accurate responses and one with the least accurate responses within the dataset. The first question pertains to the Wikipedia page of the movie *The Wonderful World of the Brothers Grimm* and the workers' responses to

this question received the highest average human evaluation score (0.959). Conversely, the workers' responses to the second question, which relates to the Wikipedia page on the "Invasion of Normandy," obtained the lowest average human evaluation score (0.34), indicating that this question was significantly more challenging to answer accurately by the workers. For each question, we provide examples of responses for low, mid-range, and high-accuracy levels.

The responses to the first question, which were mostly highly accurate (average human evaluator score: 0.959), are presented in Table O1. The lowest-quality response (Response 1) received a human evaluation score of 0 and a correspondingly low AQER score of 0.221. This was the only response that received a human evaluator score below 0.8. The subsequent responses (Responses 2–4) were scored by human evaluators between 0.8 and 0.9, with AQER scores ranging from 0.72 to 0.928. The highest-accuracy responses (Responses 5 and 6) received a human evaluation score of 1, with AQER scores between 0.975 and 0.981. These examples demonstrate AQER's strong performance when the average accuracy of workers' responses is high.

Table O1 – High, mid, and low quality responses to a question with the highest average human evaluator score

Question	Correct Answer	Response Number	Answer	Human Evaluation Score	AQER Score
According to the movie, what type of stories did Wilhelm collect? (average human evaluator score of 0.959)	Fairy tales	1	History	0	0.221
		2	Fictions, fantasies	0.8	0.725
		3	Folklore	0.9	0.720
		4	folklore, fairy stories.	0.9	0.928
		5	fairy tales	1	0.981
		6	Fairy tales.	1	0.975

The responses to the second question, which yielded low accuracy responses (average human evaluator score: 0.34), are presented in Table O2. In this table, the lowest-quality responses (Responses 1 and 2)

received a human evaluation score of 0 and correspondingly low AQER scores ranging from 0.107 to 0.187. The mid-range responses (Responses 3 and 4) were scored by human evaluators at 0.5, with AQER scores ranging from 0.32 to 0.77. The highest-accuracy responses (Responses 5 and 6) received human evaluation scores between 0.8 and 0.9, with AQER scores ranging from 0.713 to 0.75. As expected, given the low average worker score (0.34), AQER's grading of individual responses was less accurate than in table O1. However, it is evident that AQER was still able to successfully assign higher scores to the best responses (5 and 6) compared to the lowest-quality responses (1 and 2).

Table O2– High, mid, and low quality responses to a question with the lowest average human evaluator score

Question	Correct Answer	Response Number	Answer	Human Evaluation Score	AQER Score
What kind of attacks did the Allies conduct before the seaborne invasion? (average human evaluator score of 0.34)	Extensive aerial and naval bombardment and an airborne assault	1	I'm not sure	0	0.107
		2	I don't remember	0	0.187
		3	Aerial attacks	0.5	0.320
		4	Air bombs	0.5	0.77
		5	Aerial bombardment and airborne troop deployment	0.8	0.75
		6	Extensive bombardment from air and sea and airborne assault	0.9	0.713

APPENDIX P – ANALYSIS OF COSINE SIMILARITY

Table P1. presents two exemplary questions (the same ones reported in Appendix O) from our dataset, along with their correct answers. For each question, we include a selection of answers provided by workers (also the same ones reported in Appendix O), along with their respective cosine similarity scores, quantifying the semantic similarity between each worker's answer and the correct answer. These scores illustrate the

alignment between human evaluations and automatic semantic similarity measures.

Table P1 – Examples of questions, their correct answers, selected answers provided by workers, corresponding human evaluation scores, and cosine similarity scores representing the semantic similarity between the workers' answers and the correct answers.

Question	Correct answer	Answer	Human evaluation score	Cosine similarity
According to the movie, what type of stories did Wilhelm collect?	Fairy tales	History	0	0.190
		Fictions, fantasies	0.8	0.687
		Folklore	0.9	0.635
		folklore, fairy stories.	0.9	0.839
		fairy tales	1	1.0
		Fairy tales.	1	0.985
What kind of attacks did the Allies conduct before the seaborne invasion?	Extensive aerial and naval bombardment and an airborne assault	I'm not sure	0	-0.110
		I don't remember	0	-0.096
		Aerial attacks	0.5	0.479
		Air bombs	0.5	0.381
		Aerial bombardment and airborne troop deployment	0.8	0.652
		Extensive bombardment from air and sea and airborne assault	0.9	0.899

Additionally, table P2 shows the Pearson correlation coefficients between human evaluation scores and cosine similarity scores of workers' answers for each dataset, along with corresponding p-values indicating statistical significance. The correlations are positive and significant ($p < 0.001$) across all datasets.

Table P2. Pearson Correlation between human evaluation scores and cosine similarity scores of workers' answers in each dataset

Dataset	Pearson Correlation	p-value
Normandy Grimm	0.75	<0.001
Voyager IM	0.73	<0.001
CS	0.47	<0.001

REFERENCES

Androutsopoulos, Ion, and Prodromos Malakasiotis. "A survey of paraphrasing and textual entailment methods." Journal of Artificial Intelligence Research 38 (2010): 135-187.

Bajaj P, Campos D, Craswell N, Deng L, Gao J, Liu X, Majumder R et al. (2016) MS MARCO: A human generated machine reading comprehension dataset. arXiv preprint arXiv:1611.09268.

Braylan, Alexander, and Matthew Lease. "Modeling and aggregation of complex annotations via annotation distances." In Proceedings of The Web Conference 2020, pp. 1807-1818. 2020.

Chai, Lei, Hailong Sun, and Zizhe Wang. "An error consistency based approach to answer aggregation in open-ended crowdsourcing." Information Sciences 608 (2022): 1029-1044.

Cer, D., Yang, Y., Kong, S. Y., Hua, N., Limtiaco, N., John, R. S., ... & Kurzweil, R. (2018). Universal sentence encoder. arXiv preprint arXiv:1803.11175.

Fridy, J. A. (1985). On statistical convergence. Analysis, 5(4), 301-314

Gao, T., Yao, X., & Chen, D. (2021). Simcse: Simple contrastive learning of sentence embeddings. arXiv preprint arXiv:2104.08821.

Goldstein, Anat, and Chen Hajaj. "Measuring flight-destination similarity: A multidimensional approach." Expert Systems with Applications 238 (2024): 121802.

Kočiský T, Schwarz J, Blunsom P, Dyer C, Hermann KM, Melis G, Grefenstette E. (2018) The narrativeqa reading comprehension challenge. *Transactions of the Association for Computational Linguistics*. 6:317-328.

Lewis, Mike, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. "Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension." arXiv preprint arXiv:1910.13461 (2019).

Li, Jiyi. "Crowdsourced text sequence aggregation based on hybrid reliability and representation." In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1761-1764. 2020.

Li, Jiyi, and Fumiyo Fukumoto. "A dataset of crowdsourced word sequences: Collections and answer aggregation for ground truth creation." In Proceedings of the First Workshop on Aggregating and Analysing Crowdsourced Annotations for NLP, pp. 24-28. 2019.

Mohler M, Bunescu R, Mihalcea R (2011) Learning to grade short answer questions using semantic similarity measures and dependency graph alignments. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies* (ACL), 752–762.

Nguyen T, Rosenberg M, Song X, Gao J, Tiwary S, Majumder R, Deng L (2016) MS MARCO: A human generated machine reading comprehension dataset. *Workshop in Advances in Neural Information Processing Systems*. https://arxiv.org/pdf/1611.09268.pdf

Papineni K, Roukos S, Ward T, Zhu W-J (2002) Bleu: a method for automatic evaluation of machine translation. *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 311-318.

Reimers, N., & Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. arXiv preprint arXiv:1908.10084.

Roy S, Dandapat S, Nagesh A, Narahari Y (2016) Wisdom of students: A consistent automatic short answer grading technique. *Proceedings of the 13th International Conference on Natural Language Processing* (ACL), 178–187.

Rudin, W. (1976). *Principles of mathematical analysis* (3rd ed.). McGraw-Hill.

Zhang T, Kishore V, Wu F, Weinberger KQ, Artzi Y. Bertscore: Evaluating text generation with bert. arXiv preprint arXiv:1904.09675. 2019 Apr 21.

Zheng, L., Chiang, W.L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E. and Zhang, H., 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. Advances in Neural Information Processing Systems, 36, pp.46595-46623.