What Ingredients Make for an Effective Crowdsourcing Protocol for Difficult NLU Data Collection Tasks?

Nikita Nangia[1*] **Saku Sugawara**[2*] **Harsh Trivedi**[3]

Alex Warstadt[1] **Clara Vania**[4†] **Samuel R. Bowman**[1]

[1]New York University, [2]National Institute of Informatics, [3]Stony Brook University, [4]Amazon

Correspondence: {nikitanangia, bowman}@nyu.edu, saku@nii.ac.jp

Abstract

Crowdsourcing is widely used to create data for common natural language understanding tasks. Despite the importance of these datasets for measuring and refining model understanding of language, there has been little focus on the crowdsourcing methods used for collecting the datasets. In this paper, we compare the efficacy of interventions that have been proposed in prior work as ways of improving data quality. We use multiple-choice question answering as a testbed and run a randomized trial by assigning crowdworkers to write questions under one of four different data collection protocols. We find that asking workers to write explanations for their examples is an ineffective stand-alone strategy for boosting NLU example difficulty. However, we find that training crowdworkers, and then using an iterative process of collecting data, sending feedback, and qualifying workers based on expert judgments is an effective means of collecting challenging data. But using crowdsourced, instead of expert judgments, to qualify workers and send feedback does not prove to be effective. We observe that the data from the iterative protocol with expert assessments is more challenging by several measures. Notably, the human–model gap on the unanimous agreement portion of this data is, on average, twice as large as the gap for the baseline protocol data.

1 Introduction

Crowdsourcing is a scalable method for constructing examples for many natural language processing tasks. Platforms like Amazon's Mechanical Turk give researchers access to a large, diverse pool of people to employ (Howe, 2006; Snow et al., 2008; Callison-Burch, 2009). Given the ease of data collection with crowdsourcing, it has been frequently used for collecting datasets for natural language understanding (NLU) tasks like question answering (Mihaylov et al., 2018), reading comprehension (Rajpurkar et al., 2016; Huang et al., 2019), natural language inference (Dagan et al., 2005; Bowman et al., 2015; Williams et al., 2018; Nie et al., 2020a), and commonsense reasoning (Talmor et al., 2019).

There has been substantial research devoted to studying crowdsourcing methods, especially in the human-computer interaction literature (Kittur et al., 2008, 2011; Bernstein et al., 2012). However, most prior research investigates methods for collecting accurate *annotations* for existing data, for example labeling objects in images or labeling the sentiment of sentences (Hsueh et al., 2009; Liu et al., 2019a; Sun et al., 2020). There are some small-scale studies that use writing tasks, like writing product reviews, to compare crowdsourcing methodologies (Dow et al., 2012). However, we are unaware of any prior work that directly evaluates the effects of crowdsourcing protocol design choices on the quality of the resulting data for NLU tasks.

Decisions around methodology and task design used to collect datasets dictate the quality of the data collected. As models become stronger and are able to solve existing NLU datasets, we have an increasing need for difficult, high-quality datasets that are still reliably solvable by humans. As a result, our thresholds for what makes a dataset acceptable become stricter: The data needs to be challenging, have high human-agreement, and avoid serious annotation artifacts (Gururangan et al., 2018). To make collecting such large-scale datasets feasible, making well-informed crowdsourcing design decisions becomes crucial.

Existing NLP datasets have been crowdsourced with varying methods. The prevailing standard is to experiment with task design during pilots that are run before the main data collection (Vaughan, 2018). This piloting process is essential to design-

[*]Equal contribution.

[†]Work done while at New York University.

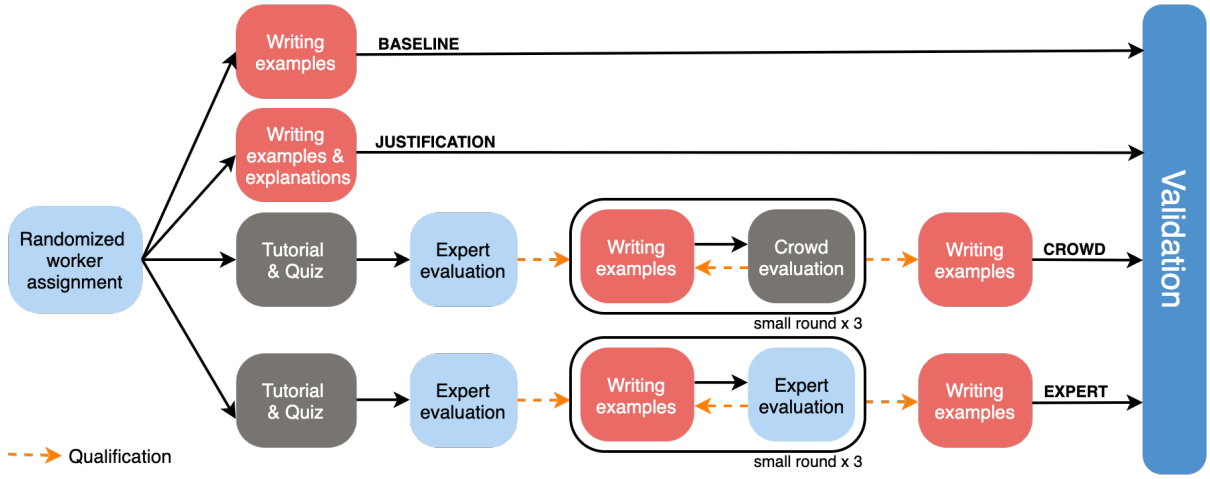

Figure 1: The initial pool of crowdworkers are randomly assigned to one of four protocols and the datasets are collected in parallel.

ing good crowdsourcing tasks with clear instructions, but the findings from these pilots are rarely discussed in published corpus papers, and the pilots are usually not large enough or systematic enough to yield definitive conclusions. In this paper, we use a randomized trial to directly compare crowdsourcing methodologies to establish general best practices for NLU data collection.

We compare the efficacy of three types of crowdsourcing interventions that have been used in previous work. We use multiple-choice question answering in English as a testbed for our study and collect four small datasets in parallel including a baseline dataset with no interventions. We choose QA as our test-bed over the similarly popular testbed task of natural language inference (NLI) because of our focus on very high human-agreement examples which calls for minimizing label ambiguity. In multiple-choice QA, the correct label is the answer choice that is *most likely to be correct*, even if there is some ambiguity in whether that choice is genuinely true . In NLI however, if more than one label is plausible, then resolving the disagreement by ranking labels may not be possible (Pavlick and Kwiatkowski, 2019). In the trial, crowdworkers are randomly assigned to one of four protocols: BASELINE, JUSTIFICATION, CROWD, or EXPERT.[1] In BASELINE, crowdworkers are simply asked to write question-answering examples. In JUSTIFICATIONthey are tasked with also writing explanations for their examples, prompting self-assessment. For the EXPERT and CROWD protocols, we train workers using an iterative process of collecting data, sending feedback, and qualifying high performing workers to subsequent rounds. We use expert-curated evaluations in EXPERT, and crowdsourced evaluations in CROWD for generating feedback and assigning qualifications. We use a a standard of high pay and strict qualifications for all protocols. We also validate the data to discard ambiguous and unanswerable examples. The experimental pipeline is sketched in Figure 1.

To quantify the dataset difficulty, we collect additional label annotations to establish human performance on each dataset and compare these to model performance. We also evaluate the difficulty of the datasets for typical machine learning models using IRT (Baker and Kim, 1993; Lalor et al., 2016).

We find that the EXPERT protocol dataset is the most challenging. The human–model gap with RoBERTa$_{\text{LARGE}}$ (Liu et al., 2019b) on the unanimous agreement portion of EXPERT is 13.9 percentage point, compared to 7.0 on the BASELINE protocol. The gap with UnifiedQA (Khashabi et al., 2020) is 6.7 on EXPERT, compared to 2.9 on BASELINE. However, the CROWD evaluation data is far less challenging than EXPERT, suggesting that expert evaluations are more reliable than crowdsourced evaluations for sending feedback and assigning qualifications.

We also find that the JUSTIFICATION intervention is ineffective as a stand-alone method for increasing NLU data quality. A substantial proportion of the explanations submitted are duplicates, reused for multiple examples, or give trivial reasoning that is not specific to the example.

[1] All the data is available at https://github.com/nyu-mll/crowdsourcing-protocol-comparison.

Lastly, to evaluate the datasets for serious annotation artifacts we test the guessability of answers by omitting the questions from the model input. This partial-input baseline achieves the lowest accuracy on EXPERT, showing that the interventions used to successfully boost example difficulty may also reduce annotation artifacts.

2 Related Work

Creating NLU Corpora Existing NLU datasets have been collected using a multitude of methods, ranging from expert-designed, to crowdsourced, to automatically scraped. The widely used Winograd schema dataset by Levesque et al. (2012) is constructed manually by specialists and it has 273 examples. Larger NLU datasets, more appropriate for training neural networks, are often crowdsourced, though the crowdsourcing methods used vary widely. Popular datasets, such as SQuAD (Rajpurkar et al., 2016) for question answering and SNLI (Bowman et al., 2015) for natural language inference, are collected by providing crowdworkers with a context passage and instructing workers to write an example given the context. Rogers et al. (2020) crowdsource QuAIL, a QA dataset, by using a more constrained data collection protocol where they require workers to write nine specific types of question for each passage. QuAC (Choi et al., 2018) is crowdsourced by pairing crowdworkers, providing one worker with a Wikipedia article, and instructing the second worker to ask questions about the hidden article.

Recently, there has been a flurry of corpora collected using adversarial models in the crowdsourcing pipeline. Dua et al. (2019), Nie et al. (2020a), and Bartolo et al. (2020) use models in the loop during data collection, where crowdworkers can only submit examples that cannot be solved by the models. However, such datasets can be biased towards quirks of the model used during data collection (Zellers et al., 2019; Gardner et al., 2020).

Crowdsourcing Methods While crowdsourcing makes it easy to collect large datasets quickly, there are some clear pitfalls: Crowdworkers are generally less knowledgeable than field experts about the requirements the data needs to meet, crowdwork can be monotonous resulting in repetitive and noisy data, and crowdsourcing platforms can create a "market for lemons" where fast work is incentivized over careful, creative work because of poor quality requesters (Akerlof, 1978; Chandler et al., 2013).

Daniel et al. (2018) give a broad overview of the variables at play when trying to crowdsource high-quality data, discussing many strategies available to requesters. Motivated by the use of self-assessment in teaching Boud (1995), Dow et al. (2012) study the effectiveness of self-assessment and external assessment when collecting data for product reviews. They find that both strategies are effective for improving the quality of submitted work. However, Gadiraju et al. (2017) find that crowdworker self-assessment can be unreliable since poor-performing workers overestimate their ability. Drapeau et al. (2016) test a justify-reconsider strategy: Crowdworkers justify their annotations in a relation extraction task, they are shown a justification written by a different crowdworker, or an expert, and are asked to reconsider their annotation. They find that this method significantly boosts the accuracy of annotations.

Another commonly used strategy when crowdsourcing NLP datasets is to only qualify workers who pass an initial quiz or perform well in preliminary crowdsourcing batches (Wang et al., 2013; Cotterell and Callison-Burch, 2014; Ning et al., 2020; Shapira et al., 2020; Roit et al., 2020). In addition to using careful qualifications, Roit et al. (2020) send workers feedback detailing errors they made in their QA-SRL annotation. Writing such feedback is labor-intensive and can become untenable as the number of workers grows. Dow et al. (2011) design a framework of promoting crowdworkers into "shepherding roles" to crowdsource such feedback. We compare expert and crowdsourced feedback in our EXPERT and CROWD protocols.

3 Data Collection Protocols

We run our study on Amazon Mechanical Turk.[2] At launch, crowdworkers are randomly assigned to one of four data collection protocols, illustrated in Figure 1.[3] To be included in the initial pool, workers need to have an approval rating of 98% or higher, have at least 1,000 approved tasks, and be located in the US, the UK, or Canada.

3.1 Writing Examples

This task is used for collecting question-answer pairs in the crowdsourcing pipeline for all four pro-

[2] https://www.mturk.com/

[3] Screenshots of the task interfaces, and code to replicate them, are provided in the git repository.

tocols. Crowdworkers assigned to the BASELINE protocol are presented with only this task.

In this writing task, we provide a context passage drawn from the Open American National Corpus (Ide and Suderman, 2006).[4] Inspired by Hu et al. (2020), we ask workers to write two questions per passage with four answer choices each. We direct workers to ensure that the questions are answerable given the passage and that there is only one correct answer for each question. We instruct them to limit word overlap between their answer choices and the passage and to write distracting answer choices that will seem plausibly correct to someone who hasn't carefully read the passage. To clarify these criteria, we provide examples of good and bad questions.

3.2 Self-Assessment

Workers assigned to the JUSTIFICATION protocol are given the writing task described above (Section 3.1) and are also tasked with writing a 1–3 sentence explanation for each question. They are asked to explain the reasoning needed to select the correct answer choice, mentioning what they think makes the question they wrote challenging.

3.3 Iterative Feedback and Qualification

Tutorial Workers assigned to the CROWD and EXPERT protocols are directed to a tutorial upon assignment. The tutorial consists of two quizzes and writing tasks. The quizzes have four steps. In each step workers are shown a passage, two question candidates and are asked to select which candidate (i) is less ambiguous, (ii) is more difficult, (iii) is more creative, or (iv) has better distracting answer choices. These concepts are informally described in the writing task instructions, but the tutorial makes the rubric explicit, giving crowdworkers a clearer understanding of our desiderata. We give workers immediate feedback on their performance during the first quiz and not the second so that we can use it for evaluation. Lastly, for the tutorial writing tasks, we provide two passages and ask workers to write two questions (with answer choices) for each passage. These questions are graded by three experts[5] using a rubric with the same metrics described in the quiz, shown in Figure 2. We give the qualification to continue onto the writing tasks to the top 60% of crowdworkers who complete the tutorial. We only qualify the workers who wrote answerable, unambiguous questions, and we qualify enough workers to ensure that we would have a large pool of people in our final writing round.

1. *Is the question answerable and unambiguous?*
   ○ Yes ○ No ○ Yes, but the label is wrong
2. *How closely do you think someone would need to read the passage to correctly answer the question?*
   ○ Wouldn't need to read it
   ○ Quickly skim a few words or one sentence
   ○ Quickly skim a few sentences
   ○ Read the whole passage
   ○ May need to read the passage more than once
3. *How creative do you think the question is?*
   ○ Not creative ○ A little creative
   ○ Fairly creative ○ Very creative
4. *Does the example have distracting answer choices?*
   ○ Yes ○ No

Figure 2: The grading rubric used to evaluate examples submitted during the intermediate writing rounds in the EXPERT and CROWD protocols.

Intermediate Writing Rounds After passing the tutorial, workers go through three small rounds of writing tasks. At the end of each round, we send them feedback and qualify a smaller pool of workers for the next round. We only collect 400–500 examples in these intermediate rounds. At the end of each round, we evaluate the submitted work using the same rubric defined in the tutorial. In the EXPERT protocol, three experts grade worker submissions, evaluating at least four questions per worker. The evaluation annotations are averaged and workers are **qualified** for the next round based on their performance. The qualifying workers are sent a message with **feedback** on their performance and a bonus for qualifying. Appendix A gives details on the feedback sent.

Evaluating the examples in each round is labor-intensive and challenging to scale (avg. 30 expert-min. per worker). In the CROWD protocol we experiment with crowdsourcing these evaluations. After the first intermediate writing round in CROWD, experts evaluate the submitted work. The evaluations are used to qualify workers for the second writing round *and* to promote the top 20% of workers into a feedback role. After intermediate writing rounds

[4] Following MultiNLI (Williams et al., 2018), we select the ten genres from OANC that are accessible to non-experts: Face-to-face, telephone, 911, travel, letters, slate, verbatim, government, OUP, and fiction.

[5] The expert annotators are authors of this paper and Dhara Mungra. All have research experience in NLU.

2 and 3, the promoted workers are tasked with evaluating all the examples (no one evaluates their own work). We collect five evaluations per example and use the averaged scores to send feedback and qualify workers for the subsequent round.

For both CROWD and EXPERT protocols, the top 80% of workers are requalified at the end of each round. Of the 150 workers who complete the tutorial, 20% qualify for the final writing round. Our qualification rate is partly dictated by a desire to have a large enough pool of people in the final writing task to ensure that no dataset is skewed by only a few people (Geva et al., 2019).

Cost We aim to ensure that our pay rate is at least US $15/hr for all tasks. The total cost per question, excluding platform fees, is $1.75 for the BASELINE protocol and $2 for JUSTIFICATION. If we discard all the data collected in the intermediate writing rounds, the cost is $3.76 per question for EXPERT,[6] and $5 for CROWD.

The average pay given during training to workers that qualify for the final writing task in EXPERT is about $120/worker (with an estimated 6–7 hours spent in training). In CROWD, there is an additional cost of $85/worker for collecting crowdsourced evaluations. The cost per example, after training, is $1.75 per question for both protocols, and total training cost does not scale linearly with dataset size, as one may not need twice as many writers for double the dataset size. More details on our payment and incentive structure can be found in Appendix B.

4 Data Validation

We collect label annotations by asking crowdworkers to pick the correct answer choice for a question, given the context passage. In addition to the answer choices written by the writer, we add an *Invalid question / No answer* option. We validate the data from each protocol. For CROWD and EXPERT, we only validate the data from the final large writing rounds. Data from all four protocols is shuffled and we run a single validation task, collecting either two or ten annotations per example.

We use the same minimum qualifications as the writing task (Section 3), and require that workers first pass a qualification task. The qualification task consists of 5 multiple-choice QA examples that have been annotated by experts.[7] People who answer at least 3 out of 5 questions correctly receive the qualification to work on the validation tasks. Of the 200 crowdworkers who complete the qualification task, 60% qualify for the main validation task. Following Ho et al. (2015), to incentivize higher quality annotations, we include expert labeled examples in the validation task, constituting 10% of all examples. If a worker's annotation accuracy on these labeled examples falls below 50%, we remove their qualification (7 workers are disqualified through this process), conversely workers who label these examples correctly receive a bonus.

10-Way Validation Pavlick and Kwiatkowski (2019) show that annotation disagreement may not be noise, but could be a signal of true ambiguity. Nie et al. (2020b) recommend using high-human-agreement data for model evaluation to avoid such ambiguity. To have enough annotations to filter the data for high human agreement and to estimate human performance, we collect ten annotations for 500 randomly sampled examples per protocol.

Cost We pay $2.50 for the qualification task and $0.75 per pair of questions for the main validation task. For every 3 out of 4 expert-labeled examples a worker annotates correctly, we send a $0.50 bonus.

5 Datasets and Analysis

We collect around 1,500 question-answer pairs from each protocol design: 1,558 for BASELINE, 1,534 for JUSTIFICATION, 1,600 for CROWD, and 1,580 for EXPERT. We use the validation annotations to determine the gold-labels and to filter out examples: If there is no majority agreement on the answer choice, or if the majority selects *invalid question*, the example is discarded ($\sim$ 5% of examples). For the 2-way annotated data, we take a majority vote over the two annotations plus the original writer's label. For the 10-way annotated data, we sample four annotations and take a majority vote over those four plus the writer's vote, reserving the remainder to compute an independent estimate of human performance.

[6] The discarded data collected during training was annotated by experts, and if we account for the cost of expert time used, the cost for EXPERT increases to $4.23/question. This estimate is based on the approximate hourly cost of paying a US PhD student, including benefits and tuition.

[7] These examples are taken from intermediate rounds 1, 2, and 3 of the EXPERT protocol.

Dataset	*N*	Human	RoBERTa	Δ	UniQA	Δ
BASELINE	*1492*	-	88.8 (0.2)	-	93.6	-
JUSTIFICATION	*1437*	-	86.5 (0.6)	-	91.4	-
CROWD	*1544*	-	81.8 (0.7)	-	88.1	-
EXPERT	*1500*	-	81.3 (0.6)	-	87.7	-
Results on the 10-way annotated subset						
BASELINE	*482*	95.9	87.2 (0.8)	8.7	92.5	3.3
JUSTIFICATION	*471*	95.5	86.7 (1.0)	8.9	90.9	**4.7**
CROWD	*472*	94.8	83.5 (1.0)	11.3	90.5	**4.3**
EXPERT	*464*	92.8	80.6 (1.1)	**12.2**	89.8	3.0
High agreement (>80%) portion of 10-way annotated data						
BASELINE	*436*	97.7	89.3 (0.8)	8.4	94.0	3.7
JUSTIFICATION	*419*	97.8	89.5 (0.6)	8.3	93.1	**4.8**
CROWD	*410*	96.8	86.2 (0.9)	10.6	93.6	3.2
EXPERT	*383*	98.2	84.7 (1.3)	**13.5**	92.9	**5.3**
Unanimous agreement portion of 10-way annotated data						
BASELINE	*340*	99.1	92.1 (0.7)	7.0	96.2	2.9
JUSTIFICATION	*307*	98.7	93.2 (0.3)	5.5	95.8	2.9
CROWD	*277*	98.6	88.9 (0.9)	9.7	97.1	1.4
EXPERT	*271*	99.3	85.4 (1.1)	**13.9**	92.5	**6.7**

Table 1: Human and model performance on each of our datatsets. *N* shows the number of examples in the dataset. *RoBERTa* shows average zero-shot performance for six RoBERTa$_{\text{LARGE}}$ models finetuned on RACE, standard deviation is in parentheses. *UniQA* shows zero-shot performance of the T5-based UnifiedQA-v2 model. Δ shows the differences in human and model performance.

5.1 Human Performance and Agreement

For the 10-way annotated subsets of the data, we take a majority vote over the six annotations that are *not* used when determining the gold answer, and compare the result to the gold answer to estimate human performance. Table 1 shows the result for each dataset. The EXPERT and CROWD datasets have lower human performance numbers than BASELINE and JUSTIFICATION. This is also mirrored in the inter-annotator agreement for validation, where Krippendorf's α (Krippendorff, 1980) is 0.67 and 0.71 for EXPERT and CROWD, compared to 0.81 and 0.77 for BASELINE and JUSTIFICATION (Table 3 in Appendix C). The lower agreement may be reflective of the fact that while these examples are still clearly human solvable, they are more challenging than those in BASELINE and JUSTIFICATION As a result, annotators are prone to higher error rates, motivating us to look at the higher agreement portions of the data to determine true dataset difficulty. And while the agreement rate is lower for EXPERT and CROWD, more than 80% of the data still has high human-agreement on the gold-label, where at least 4 out of 5 annotators agree on the label. The remaining low-agreement examples may have more ambiguous questions, and we follow Nie et al.'s (2020b) recommendation and focus our analysis on the high-agreement portions of the dataset.

5.2 Zero-Shot Model Performance

We test two pretrained models that perform well on other comparable QA datasets: RoBERTa$_{\text{LARGE}}$ (Liu et al., 2019b) and UnifiedQA-v2 (Khashabi et al., 2020). We fine-tune RoBERTa$_{\text{LARGE}}$ on RACE (Lai et al., 2017), a large-scale multiple-choice QA dataset that is commonly used for training (Sun et al., 2019). We fine-tune 6 RoBERTa$_{\text{LARGE}}$ models and report the average performance across runs. The UnifiedQA-v2 model is a single T5-based model that has been trained on 15 QA datasets.[8] We also fine-tune RoBERTa$_{\text{LARGE}}$ on CosmosQA and QuAIL, finding that zero-shot model performance is best with RACE fine-tuning but that the trends in model accuracy across our four datasets are consistent (Appendix D).

5.3 Comparing Protocols

As shown in Table 1, model accuracy on the full datasets is lowest for EXPERT, followed by CROWD, JUSTIFICATION, and then BASELINE. However, model accuracy alone does not tell us how much

[8]The authors of UnifiedQA kindly shared the unreleased v2 model with us.

headroom is left in the datasets. Instead, we look at the difference between the estimated human performance and model performance.

Human–Model gap The trends in the human–model gap on the 10-way annotated sample are inconsistent across models. For a more conclusive analysis, we focus on the higher-agreement portions of the data where label ambiguity is minimal.

On the high agreement section of the datasets, both models' performance is weakest on EXPERT. RoBERTa$_{\text{LARGE}}$ shows the second largest human–model gap on CROWD, however for UnifiedQA JUSTIFICATION is the next hardest dataset. This discrepancy between the two types of iterative feedback protocols is even more apparent in the unanimous agreement portion of the data. On the unanimous agreement examples, both models show the lowest performance on EXPERT but UnifiedQA achieves near perfect performance on CROWD. This suggests that while the CROWD protocol used nearly the same crowdsourcing pipeline as EXPERT, the evaluations done by experts are a much more reliable metric for selecting workers to qualify and for generating feedback, at the cost of greater difficulty with scaling to larger worker pools. This is confirmed by inter-annotator agreement: Expert agreement on the rubric-based evaluations has a Krippendorf's α of 0.65, while agreement between crowdworker evaluations is 0.33.

Self-Justification Model performance on the unanimous agreement examples of JUSTIFICATION is comparable to, or better than, performance on BASELINE. To estimate the quality of justifications, we manually annotate a random sample of 100 justifications. About 48% (95% CI: [38%, 58%]) are duplicates or near-duplicates of other justifications, and of this group, nearly all are trivial (e.g. *Good and deep knowledge is needed to answer this question*) and over half are in non-fluent English (e.g. *To read the complete passage to understand the question to answer.*). On the other hand, non-duplicate justifications are generally of much higher quality, mentioning distractors, giving specific reasoning, and rewording phrases from the passage (e.g. *Only #1 is discussed in that last paragraph. The rest of the parts are from the book, not the essay. Also the answer is paraphrased from "zero-sum" to "one's gain is another's loss"*). While we find that JUSTIFICATION does not work as a stand-alone strategy, we cannot conclude that self-justification would be equally ineffective if combined with more aggressive screening to exclude crowdworkers who author trivial or duplicate justifications. Gadiraju et al. (2017) also recommend using the accuracy of a worker's self-assessments to screen workers.

Partial input	P + A	Q + A	A
BASELINE	69.9 (4.7)	41.9 (2.9)	34.9 (2.4)
JUSTIFICATION	57.9 (1.3)	38.3 (2.2)	33.9 (6.3)
CROWD	57.7 (3.1)	43.9 (2.0)	35.2 (1.9)
EXPERT	52.0 (1.5)	42.8 (1.8)	35.7 (1.4)

Table 2: Accuracy (std.) of partial input baselines. *P* is passage, *Q* is question, and *A* is answer choices.

Cross-Protocol Transfer Since the datasets from some protocols are clearly more challenging than others, it prompts the question: are these datasets also better for training models? To test cross-protocol transfer, we fine-tune RoBERTa$_{\text{LARGE}}$ on one dataset and evaluate on the other three. We find that model accuracy is not substantively better from fine-tuning on any one dataset (Table 5, Appendix E). The benefit of EXPERT being a more challenging evaluation dataset does not clearly translate to training. However, these datasets may be too small to offer clear and distinguishable value in this setting.

Annotation Artifacts To test for undesirable artifacts, we evaluate partial input baselines (Kaushik and Lipton, 2018; Poliak et al., 2018). We take a RoBERTa$_{\text{LARGE}}$ model, pretrained on RACE, and fine-tune it using five-fold cross-validation, providing only part of the example input. We evaluate three baselines: providing the model with the passage and answer choices only, the question and answer choices only, and the answer choices alone. Results are shown in Table 2. The passage+answer baseline has significantly lower performance on the EXPERT dataset in comparison to the others. This indicates that the iterative feedback and qualification method using expert assessments not only increases overall example difficulty but may also lower the prevalence of simple artifacts that can reveal the answer. Performance of the question+answer and answer-only baselines is comparably low on all four datasets.

Question and Answer Length We observe that the difficulty of the datasets is correlated with average answer length (Figure 3). The hardest dataset, EXPERT, also has the longest answer options with

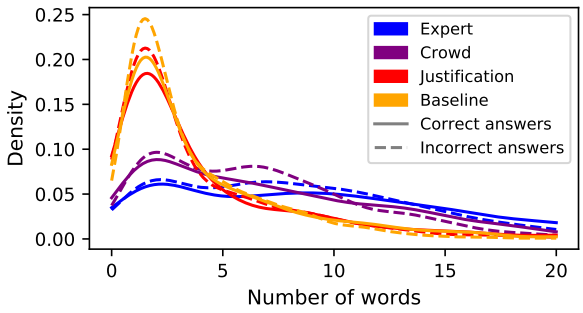

Figure 3: Distribution of answer lengths. The distributions for different datasets and for the correct and incorrect answer options are plotted separately.

an average of 9.1 words, compared to 3.7 for BASELINE, 4.1 for JUSTIFICATION, and 6.9 for CROWD. This reflects the tendency of the 1- and 2-word answers common in the BASELINE and JUSTIFICATION datasets to be extracted directly from the passage. While sentence-length answers, more common in EXPERT and CROWD, tend to be more abstractive. Figure 3 also shows that incorrect answer options tend to be shorter than correct ones. This pattern holds across all datasets, suggesting a weak surface cue that models could exploit. Using an answer-length based heuristic alone, accuracy is similar to the answer-only model baseline: 34.2% for BASELINE, 31.7% for JUSTIFICATION, 31.5% for CROWD, and 34.3% for EXPERT.

Wh-words We find that the questions in EXPERT and CROWD protocols have similar distributions of wh-words, with many *why* questions and few *who* or *when* questions compared to the BASELINE and JUSTIFICATION protocols, seemingly indicating that this additional feedback prompts workers to write more complex questions.

Non-Passage-Specific Questions We also observe that many questions in the datasets are formulaic and include no passage-specific content, for instance *Which of the following is true?*, *What is the main point of the passage?*, and *Which of the following is not mentioned in the passage?*. We manually annotate 200 questions from each protocol for questions of this kind. We find that there is no clear association between the dataset's difficulty and the frequency of such questions: 15% of questions in EXPERT are generic, compared to 4% for CROWD, 10% for JUSTIFICATION, and 3% for BASELINE. We might expect that higher quality examples that require reading a passage closely would ask questions that are specific rather than generic. But our results suggest that difficulty may be due more to the subtlety of the answer options, and the presence of distracting options, rather than the complexity or originality of the questions.

Order of Questions We elicit two questions per passage in all four protocols with the hypothesis that the second question may be more difficult on aggregate. However, we find that there is only a slight drop in model accuracy from the first to second question on the CROWD and EXPERT datasets (1.0 and 0.7 percentage points). And model accuracy on BASELINE remains stable, while it increases by 2.7 percentage points on JUSTIFICATION. A task design with minimal constraints, like ours, does not prompt workers to write an easier question followed by a more difficult one, or vice versa.

5.4 Item Response Theory

Individual examples within any dataset can have different levels of difficulty. To better understand the distribution of difficult examples in each protocol, we turn to Item Response Theory (IRT; Baker and Kim, 1993), which has been used to estimate individual example difficulty based on model responses (Lalor et al., 2019; Martínez-Plumed et al., 2019). Specifically, we use the three-parameter logistic (3PL) IRT model, where an example is characterized by discrimination, difficulty, and guessing parameters. Discrimination defines how effective an example is at distinguishing between weak and strong models, difficulty defines the minimum ability of a model needed to obtain high performance, and the guessing parameter defines the probability of a correct answer by random guessing. Following Vania et al. (2021), we use 90 Transformer-based models fine-tuned on RACE, with varying ability levels, and use their predictions on our four datasets as responses. For comparison, we also use model predictions on QuAIL and CosmosQA. Refer to Appendix F for more details.

Figure 4 shows the distribution of example difficulty for each protocol. Also plotted are the difficulty parameters for the intermediate rounds of data that are collected in the iterative feedback protocols.[9] We see that EXPERT examples have the highest median and 75th percentile difficulty scores,

[9]The IRT parameters for discrimination range from 0.6 to 2.1, while for guessing they range from 0.03 to 0.74. However, we observe that the distributions of both parameters across the four datasets are similar.

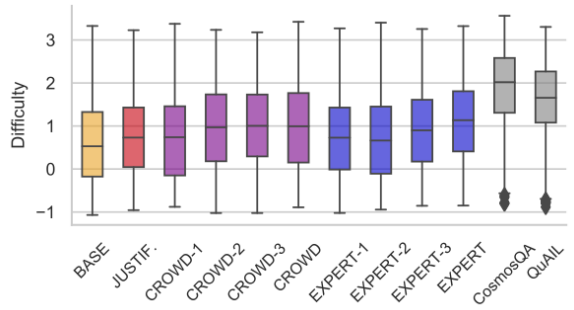

Figure 4: Distribution of examples according to their difficulty parameters. CROWD/EXPERT-$\{1, 2, 3\}$ are the three intermediate rounds of data that are not included in the final datasets.

while BASELINE scores the lowest. We also note that the greatest gain in difficulty for CROWD examples happens between rounds 1 and 2, the only feedback and qualification stage that is conducted by experts. This offers further evidence that expert assessments are more reliable, and that crowdsourcing such assessments poses a significant challenge.

While the examples in EXPERT have higher difficulty scores than the other protocols, the scores are significantly lower than those for CosmosQA and QuAIL (all four datasets show similar discrimination scores to CosmosQA and QuAIL). The data collection methods used for both CosmosQA and QuAIL differ substantially from methods we tested. Rogers et al. (2020) constrain the task design for QuAIL and require workers to write questions of specific types, like those targeting temporal reasoning. Similarly, in CosmosQA workers are encouraged to write questions that require causal or deductive commonsense reasoning. In contrast, we avoid dictating question type in our instructions. The IRT results here suggest that using prior knowledge to slightly constrain the task design can be effective for boosting example difficulty. In addition to differing task design, CosmosQA and QuAIL also use qualitatively different sources for passages. Both datasets use blogs and personal stories, QuAIL also uses texts from published fiction and news. Exploring the effect of source text genre on crowdsourced data quality is left to future work.

6 Conclusion

We present a study to determine effective protocols for crowdsourcing difficult NLU data. We run a randomized trial to compare interventions in the crowdsourcing pipeline and task design. Our results suggest that asking workers to write justifications is not a helpful stand-alone strategy for improving NLU dataset difficulty, at least in the absence of explicit incentives for workers to write high-quality justifications. However, we find that training workers using an iterative feedback and requalification protocol is an effective strategy for collecting high-quality QA data. The benefit of this method is most evident in the high-agreement subset of the data where label noise is low. We find that using expert assessments to conduct this iterative protocol is fruitful, in contrast with crowdsourced assessments that have much lower inter-annotator agreement and the noisy signal from these assessments does not boost example difficulty.

Acknowledgements

We thank Dhara Mungra for her early contributions to this project, and for being one of the expert graders during data collection. We also thank Daniel Khashabi for giving us access to UnifiedQA-v2 for our experiments. This work has benefited from financial support to SB by Eric and Wendy Schmidt (made by recommendation of the Schmidt Futures program), Apple, and Intuit, and from in-kind support by the NYU High-Performance Computing Center and by NVIDIA Corporation (with the donation of a Titan V GPU). SS was supported by JST PRESTO Grant No. JPMJPR20C4. This material is based upon work supported by the National Science Foundation under Grant No. 1922658. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Ethics Statement

We are cognizant of the asymmetrical relationship between requesters and workers in crowdsourcing, and we take care to be responsive employers and to pay a wage commensurate with the high-quality work we're looking for. So in additional to the ethical reasons for paying fair wages, our successes with collecting high-quality NLU data offer weak evidence that others should also follow this practice. However, the mere existence of more research on NLU crowdsourcing with positive results could arguably encourage more people to do crowdsourcing under a conventional model, with low pay and little worker recourse against employer malpractice. The only personal information we collect from workers

is their Mechanical Turk worker IDs, which we keep secure and will not release. However, we do not engage with issues of bias during data collection and we expect that the data collected under all our protocols will, at least indirectly, reinforce stereotypes.

We confirmed with New York University's IRB that crowdsourced NLP dataset construction work, *including* experimental work on data collection methods, is exempt from their oversight. The only personal information we collect from workers is their Mechanical Turk worker IDs, which we keep secure and will not release.

References

George A. Akerlof. 1978. The market for "lemons": Quality uncertainty and the market mechanism. In *Uncertainty in Economics*. Academic Press.

Frank B. Baker and Seock-Ho Kim. 1993. Item response theory: Parameter estimation techniques. *Journal of the American Statistical Association*, 88:707–707.

Max Bartolo, Alastair Roberts, Johannes Welbl, Sebastian Riedel, and Pontus Stenetorp. 2020. Beat the AI: Investigating adversarial human annotation for reading comprehension. *Transactions of the Association for Computational Linguistics*, 8:662–678.

Michael S. Bernstein, D. Karger, R. Miller, and J. Brandt. 2012. Analytic methods for optimizing realtime crowdsourcing. arXiv preprint 1204.2995.

David Boud. 1995. *Enhancing Learning Through Self-Assessment*. Philadelphia: Kogan Page.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Chris Callison-Burch. 2009. Fast, cheap, and creative: Evaluating translation quality using Amazon's Mechanical Turk. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 286–295, Singapore. Association for Computational Linguistics.

Jesse Chandler, Gabriele Paolacci, and Pam Mueller. 2013. Risks and rewards of crowdsourcing marketplaces. In Pietro Michelucci, editor, *Handbook of Human Computation*, pages 377–392. Springer New York.

Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wentau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. QuAC: Question answering in context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2174–2184, Brussels, Belgium. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Ryan Cotterell and Chris Callison-Burch. 2014. A multi-dialect, multi-genre corpus of informal written Arabic. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 241–245, Reykjavik, Iceland. European Language Resources Association (ELRA).

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The PASCAL recognising textual entailment challenge. In *Machine Learning Challenges Workshop*, pages 177–190. Springer.

Florian Daniel, Pavel Kucherbaev, Cinzia Cappiello, Boualem Benatallah, and Mohammad Allahbakhsh. 2018. Quality control in crowdsourcing: A survey of quality attributes, assessment techniques, and assurance actions. *ACM Comput. Surv.*, 51(1).

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Steven Dow, Anand Kulkarni, Brie Bunge, Truc Nguyen, Scott Klemmer, and Björn Hartmann. 2011. Shepherding the crowd: Managing and providing feedback to crowd workers. In *CHI '11 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '11, page 1669–1674, New York, NY, USA. Association for Computing Machinery.

Steven Dow, Anand Kulkarni, Scott Klemmer, and Björn Hartmann. 2012. Shepherding the crowd yields better work. In *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work*, CSCW '12, page 1013–1022, New York, NY, USA. Association for Computing Machinery.

Ryan Drapeau, L. Chilton, Jonathan Bragg, and Daniel S. Weld. 2016. MicroTalk: Using argumentation to improve crowdsourcing accuracy. In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing (HCOMP)*, pages 32–41. AAAI Press.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2368–2378, Minneapolis, Minnesota. Association for Computational Linguistics.

Ujwal Gadiraju, Besnik Fetahu, Ricardo Kawase, Patrick Siehndel, and Stefan Dietze. 2017. Using worker self-assessments for competence-based preselection in crowdsourcing microtasks. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 24(4):1–26.

Matt Gardner, Yoav Artzi, Victoria Basmov, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, Nitish Gupta, Hannaneh Hajishirzi, Gabriel Ilharco, Daniel Khashabi, Kevin Lin, Jiangming Liu, Nelson F. Liu, Phoebe Mulcaire, Qiang Ning, Sameer Singh, Noah A. Smith, Sanjay Subramanian, Reut Tsarfaty, Eric Wallace, Ally Zhang, and Ben Zhou. 2020. Evaluating models' local decision boundaries via contrast sets. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1307–1323, Online. Association for Computational Linguistics.

Mor Geva, Yoav Goldberg, and Jonathan Berant. 2019. Are we modeling the task or the annotator? an investigation of annotator bias in natural language understanding datasets. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1161–1166, Hong Kong, China. Association for Computational Linguistics.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112, New Orleans, Louisiana. Association for Computational Linguistics.

Chien-Ju Ho, Aleksandrs Slivkins, Siddharth Suri, and Jennifer Wortman Vaughan. 2015. *Incentivizing High Quality Crowdwork*, page 419–429. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE.

Jeff Howe. 2006. The rise of crowdsourcing. *Wired magazine*, 14(6):1–4.

Pei-Yun Hsueh, Prem Melville, and Vikas Sindhwani. 2009. Data quality from crowdsourcing: A study of annotation selection criteria. In *Proceedings of the NAACL HLT 2009 Workshop on Active Learning for Natural Language Processing*, pages 27–35, Boulder, Colorado. Association for Computational Linguistics.

Hai Hu, Kyle Richardson, Liang Xu, Lu Li, Sandra Kübler, and Lawrence Moss. 2020. OCNLI: Original Chinese Natural Language Inference. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3512–3526, Online. Association for Computational Linguistics.

Lifu Huang, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Cosmos QA: Machine reading comprehension with contextual commonsense reasoning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2391–2401, Hong Kong, China. Association for Computational Linguistics.

Nancy Ide and Keith Suderman. 2006. Integrating linguistic resources: The American national corpus model. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, Genoa, Italy. European Language Resources Association (ELRA).

Divyansh Kaushik and Zachary C. Lipton. 2018. How much reading does reading comprehension require? a critical investigation of popular benchmarks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5010–5015, Brussels, Belgium. Association for Computational Linguistics.

Daniel Khashabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. 2020. UNIFIEDQA: Crossing format boundaries with a single QA system. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1896–1907, Online. Association for Computational Linguistics.

Aniket Kittur, Ed H. Chi, and Bongwon Suh. 2008. Crowdsourcing user studies with mechanical turk. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, page 453–456, New York, NY, USA. Association for Computing Machinery.

Aniket Kittur, Boris Smus, Susheel Khamkar, and Robert E. Kraut. 2011. Crowdforge: Crowdsourcing complex work. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, UIST '11, page 43–52, New York, NY, USA. Association for Computing Machinery.

Klaus Krippendorff. 1980. *Content analysis: An introduction to its methodology*. Beverly Hills, CA: Sage publications.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. RACE: Large-scale ReAding comprehension dataset from examinations. In

Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 785–794, Copenhagen, Denmark. Association for Computational Linguistics.

John P. Lalor, Hao Wu, and Hong Yu. 2016. Building an evaluation scale using item response theory. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 648–657, Austin, Texas. Association for Computational Linguistics.

John P. Lalor, Hao Wu, and Hong Yu. 2019. Learning latent parameters without human response patterns: Item response theory with artificial crowds. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4249–4259, Hong Kong, China. Association for Computational Linguistics.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *8th International Conference on Learning Representations, ICLR 2020*.

Hector J Levesque, Ernest Davis, and Leora Morgenstern. 2012. The Winograd schema challenge. In *Proceedings of the Thirteenth International Conference on Principles of Knowledge Representation and Reasoning*, pages 552–561.

Shixia Liu, Changjian Chen, Yafeng Lu, Fangxin Ouyang, and Bin Wang. 2019a. An interactive method to improve crowdsourced annotations. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):235–245.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. RoBERTa: A robustly optimized BERT pretraining approach. arXiv preprint 1907.11692.

Fernando Martínez-Plumed, Ricardo B.C. Prudêncio, Adolfo Martínez-Usó, and José Hernández-Orallo. 2019. Item response theory in AI: Analysing machine learning classifiers at the instance level. *Artificial Intelligence*, 271:18 – 42.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391, Brussels, Belgium. Association for Computational Linguistics.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020a. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901, Online. Association for Computational Linguistics.

Yixin Nie, Xiang Zhou, and Mohit Bansal. 2020b. What can we learn from collective human opinions on natural language inference data? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9131–9143, Online. Association for Computational Linguistics.

Qiang Ning, Hao Wu, Rujun Han, Nanyun Peng, Matt Gardner, and Dan Roth. 2020. TORQUE: A reading comprehension dataset of temporal ordering questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1158–1172, Online. Association for Computational Linguistics.

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

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. 2018. Hypothesis only baselines in natural language inference. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 180–191, New Orleans, Louisiana. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Anna Rogers, Olga Kovaleva, Matthew Downey, and Anna Rumshisky. 2020. Getting closer to AI complete question answering: A set of prerequisite real tasks. In *Proceedings of The Thirty-Fourth AAAI Conference on Artificial Intelligence*, pages 8722–8731. AAAI Press.

Paul Roit, Ayal Klein, Daniela Stepanov, Jonathan Mamou, Julian Michael, Gabriel Stanovsky, Luke Zettlemoyer, and Ido Dagan. 2020. Controlled crowdsourcing for high-quality QA-SRL annotation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7008–7013, Online. Association for Computational Linguistics.

Ori Shapira, Ramakanth Pasunuru, Hadar Ronen, Mohit Bansal, Yael Amsterdamer, and Ido Dagan. 2020. Evaluating interactive summarization: an expansion-based framework. arXiv preprint 2009.08380.

Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Ng. 2008. Cheap and fast – but is it good? evaluating non-expert annotations for natural language tasks. In *Proceedings of the 2008 Conference*

on Empirical Methods in Natural Language Processing, pages 254–263, Honolulu, Hawaii. Association for Computational Linguistics.

David Q. Sun, Hadas Kotek, Christopher Klein, Mayank Gupta, William Li, and Jason D. Williams. 2020. Improving human-labeled data through dynamic automatic conflict resolution. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3547–3557, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Kai Sun, Dian Yu, Dong Yu, and Claire Cardie. 2019. Improving machine reading comprehension with general reading strategies. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2633–2643, Minneapolis, Minnesota. Association for Computational Linguistics.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

Clara Vania, Phu Mon Htut, William Huang, Dhara Mungra, Richard Yuanzhe Pang, Jason Phang, Haokun Liu, Kyunghyun Cho, and Samuel R. Bowman. 2021. Comparing test sets with item response theory. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Jennifer Wortman Vaughan. 2018. Making better use of the crowd: How crowdsourcing can advance machine learning research. *Journal of Machine Learning Research*, 18(193):1–46.

Aobo Wang, Cong Duy Vu Hoang, and Min-Yen Kan. 2013. Perspectives on crowdsourcing annotations for natural language processing. *Language Resources and Evaluation*, 47:9–31.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.

Yian Zhang, Alex Warstadt, Haau-Sing Li, and Samuel R. Bowman. 2021. When do you need billions of words of pretraining data? In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

A Iterative Protocol Feedback

In the EXPERT and CROWD protocols, we conduct three small intermediate rounds of data collection to help train crowdworkers and give them feedback on their submissions. At the end of each small round of writing, the submitted examples are evaluated either by experts or crowdworkers, as described in Section 3.3. The rubric given in Figure 2 is used during evaluations. After compiling the evaluations, we qualify the top 80% of workers for the next round and send them a feedback message. We tell workers what their difficulty and creativity scores are in comparison to the average. We also tell them what percentage of their question-answer pairs were labeled as having distracting answer choices and what percentage were labeled ambiguous, with examples of any such questions. Lastly, we list the examples they wrote that received the highest and lowest overall rubric scores.

B Payment and Incentive Structure

The compensation for for writing two questions in the baseline writing task is $3.50, excluding platform fees, we estimate it takes 12–15 minutes to do a close reading of the passage and write two challenging questions. For the JUSTIFICATION protocol, the compensation is $4 per task to account for the additional time it takes to write a justifications for each question. For the tutorial that workers in the CROWD and EXPERT protocols need to complete, we pay $3.50, and give a bonus of $1.50 if they qualify onto the writing tasks. Similarly, at the end of each intermediate writing batch, a bonus is sent to the workers that qualify for the subsequent round: $5, $7, and $10 after the 1st, 2nd and 3rd rounds respectively. Promoted workers who are tasked with the crowdsourced evaluations in the CROWD protocol, are paid $0.50 per question. They are also sent a bonus of $5 for each round of evaluations they complete.

C Inter-Annotator Agreement

Table 3 shows the inter-annotator agreement during data validation task for each dataset. The Krippendorf's α is lowest for EXPERT, which also has the lowest human performance baseline, likely due to the pressure to produce subtle questions.

Protocol	α_{all}	α_{10}
BASELINE	0.81	0.79
JUSTIFICATION	0.77	0.74
CROWD	0.71	0.69
EXPERT	0.67	0.64

Table 3: Inter-annotator agreement statistics for each datatset. α_{all} and α_{10} give the Krippendorf's α scores for all examples and the subset of 10-way annotated examples respectively.

Dataset	RACE	CosmosQA	QuAIL
BASELINE	88.8	74.1	80.5
JUSTIFICATION	86.5	65.9	68.8
CROWD	81.8	65.1	62.7
EXPERT	81.3	56.8	52.4

Table 4: Zero-shot model accuracy on our datasets, when training on the datasets named in the columns.

D Zero-Shot Model Performance: CosmosQA and QuAIL

In addition to fine-tuning RoBERTa$_{\text{LARGE}}$ on RACE, we also fine-tune it on CosmosQA, and QuAIL to test zero-shot model performance. Table 4 shows the zero-shot results. We observe that model performance on our datasets is substantially worse when fine-tuning on CosmosQA or QuAIL. However, the pattern in model behaviour is consistent regardless of corpus used. In all three conditions, model accuracy is highest on BASELINE, followed by JUSTIFICATION, then CROWD, and finally EXPERT.

E Cross-Protocol Transfer

As discussed in Section 5.3, we test cross-protocol transfer by fine-tuning RoBERTa$_{\text{LARGE}}$ on one dataset and evaluating on the other three. For a baseline comparison, we also fine-tune the model on each dataset using five-fold cross-validation. Results are shown in Table 5.

	BASE	JUST	CROWD	EXP	Cross-val
BASE	-	88.2	87.4	87.8	87.9 (2.0)
JUST	84.9	-	85.3	84.9	85.6 (2.4)
CROWD	81.6	83.2	-	81.7	82.5 (1.9)
EXPERT	80.6	81.2	81.7	-	82.8 (1.4)

Table 5: Cross-protocol evaluation where the row and column indicate target and source datasets respectively. *Cross-val* shows the accuracy and std. dev. from five-fold cross-validation on each dataset.

F IRT Setup

IRT Model We use the 3PL IRT model, where the probability of a responder i of answering an item j is given as:

$$p_j(\theta_i) = \gamma_j + \frac{1 - \gamma_j}{1 + e^{-\alpha_j(\theta_i - \beta_j))}}$$

where α, β, γ denote the discrimination, the difficulty, and the guessing parameters, respectively. Following Lalor et al. (2019), we use variational inference (VI) to estimate these parameters. Given a set of model responses M, we use the following variational posterior to estimate the joint probability of the parameters $\pi(\theta, \alpha, \beta, \gamma \mid M)$:

$$q(\theta, \alpha, \beta, \gamma) = \prod_{i=1}^{I} \pi_i^{\theta}(\theta_i) \prod_{j=1}^{J} \pi_j^{\alpha}(\alpha_i) \pi_j^{\beta}(\beta_i) \pi_j^{\gamma}(\gamma_i),$$

where $\pi^{\rho}(\cdot)$ is the density for parameter ρ. We use the following distributions for each parameter: $\mathcal{N}(\mu_\theta, \sigma_\theta^2)$ for θ, $\mathcal{N}(\mu_\alpha, \sigma_\alpha^2)$ for $\log \alpha$, $\mathcal{N}(\mu_\beta, \sigma_\beta^2)$ for β, and $\mathcal{N}(\mu_\gamma, \sigma_\gamma^2)$ for $\text{sigmoid}^{-1}(\gamma)$. We then fit the posterior parameters by minimizing the KL divergence between $q(\theta, \alpha, \beta, \gamma)$ and the true posterior $\pi(\theta, \alpha, \beta, \gamma \mid Y)$. This is equivalent to minimizing the evidence lower bound (ELBO).

To control for different test sizes, we weight the log likelihood of each item's parameter by the inverse of the item's test size when fitting the parameters. We adapt prior used by Lalor et al. (2019) for each parameter: $\mathcal{N}(0, 1)$ for θ, β, and $\text{sigmoid}^{-1}(\gamma)$. For $\log \alpha$, we use $\mathcal{N}(0, \sigma_\alpha^2)$ where we set σ_α by searching $[0.25, 0.5]$ by increments of 0.05 and use the value yielding the highest ELBO.

Pretrained Transformer Models We use 18 Transformer-based models: ALBERT-XXL-v2 (Lan et al., 2020), RoBERTa$_{\text{LARGE}}$ and RoBERTa$_{\text{BASE}}$ (Liu et al., 2019b), BERT$_{\text{LARGE}}$ and BERT$_{\text{BASE}}$ (Devlin et al., 2019), XLM-R (Conneau et al., 2020), and 12 MiniBERTas (Zhang et al., 2021).[10] We fine-tune each of these models on RACE, and keep five different checkpoints—at 1%, 10%, 25%, and 50% of the maximum training epochs, plus the best checkpoint on the RACE validation set. In total, we have 90 model responses for each test example. For all the models, we use a batch size of 8, learning rate of 1.0×10^{-5}, and finetune the models using the Adam optimizer for 4 epochs on the RACE dataset.

[10] We use pretrained models distributed with HuggingFace Transformers (Wolf et al., 2020).