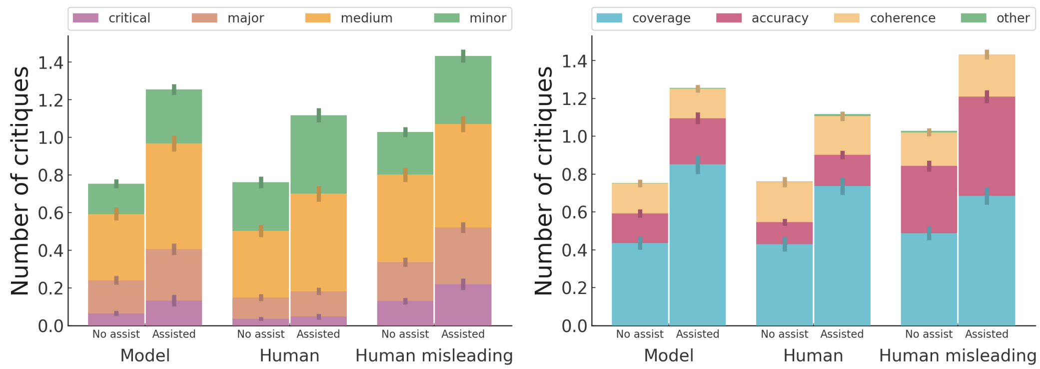

Figure 2: Even though the largest contribution from our models is finding more minor errors, model assistance also helps uncover some major and critical errors (left). The majority of model-written critiques point out lack of summary coverage, but there is also increase in found inaccuracies (right).

Importantly, our models do not have to always produce valid critiques to be helpful to human labelers, though too many invalid critiques would waste the labelers' time.

3.2 Setup

Labelers are given a topic-based summarization question and summary, and instructed to write as many different possible critiques as they can think of for the summary. They are also instructed to only read as much of the text as necessary to write the critiques.

In the baseline **no assist** condition, the labeler had no assistance and had to write all critiques themselves. In the **assist** condition, the labeler would also see 8 model-generated critiques of the answer, one at each of the following temperatures: [0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 1.0]. Critiques were de-duplicated, sorted according to the model's helpfulness score, and shown to the human labeler along with their helpfulness scores. The labeler could click a button to "use" any of the model generated critiques. If they did this, it would pre-fill the labeler critique with the model-written text, which they could further edit. We use our best model for generating all assistance.

We use questions for longer passages (2048 to 3840 tokens in length) in order to observe a larger effect from assistance. We noticed no degradation in critique writing ability despite training only on passages up to length 2048.

For each question, we use three different kinds of answers to critique:

- **Model**: An answer sampled from our best model, the same as the critiquing model, using a randomly chosen temperature.[3]
- **Human**: An answer generated by a labeler doing the base task honestly.
- **Human misleading**: An answer generated by the same labeler to deliberately have a hard-to-detect flaw. (Details in Appendix A).

This results in six total conditions. For each summary, we had six labelers finding critiques, assigning one to each of the six conditions. We ensured labelers had not seen the text before (if they had, it would get reassigned). We did not limit the amount of time the labeler took performing the task: we instructed them to take enough time to cover all possible critiques of the answer at least 90% of the time.

We also ask the labeler to provide a 1-7 Likert rating for quality of the answer, with 7 indicating a perfect answer. For each critique the labeler finds, we ask for a severity level, ranging from minor to critical, and a category (described in Appendix A).

[3]Throughout the paper, "randomly chosen temperature" means a temperature drawn uniformly at random from the set $\{0, 0.1, \ldots, 0.9, 1\}$.

3.3 Findings

We find that critiques written by our model significantly help labelers find critiques:

1. **Assistance helps labelers find more critiques under all answer conditions (Figure 1).** In the assistance conditions, most labeler-written critiques "used" a model-critique and the number of used critiques is comparable to the number of critiques found without assistance. Only 15.3% of the used critiques were edited further.
2. **Assistance does not appreciably affect the time taken by labelers (Figure 15).** The primary effect is that labelers take a bit longer because they write more critiques when given assistance.
3. **The number of critiques increases for all severity levels and critique categories (Figure 2)**. Though the biggest increases comes from minor critiques, we also find increases in major and critical critiques. Similarly, we see the biggest increase in coverage critiques, but also a modest increase in accuracy critiques.
4. **With assistance answers are more frequently critiqued and are given lower ratings (Figure 16)**. This suggests that the increase in critiques translates to a different subjective sense of quality. For example, on model answers, unassisted labelers found critiques on just 54% of inputs and gave an average rating of 5.6, while assisted labelers found critiques on 78% of inputs and gave an average rating of 4.9.

Note that our critique model was trained on data from the same pool of labelers—we are essentially leveraging our models to bootstrap our labelers to be more careful at the base task.

3.3.1 Checking for intended critiques

One potential confounder for our results is that because labelers are shown many critiques, they are occasionally erring and accepting critiques that are not truly valid. In other words, we have no way to distinguish between a valid critique of a poor answer from a misleading critique of a valid answer.

To break this symmetry, we had the authors of the misleading answers also generate the corresponding "intended" critiques (usually just one per answer). We then ask them to check whether the other labelers in the assistance experiment found critiques covering the intended critiques. We also ask them to independently judge whether each of the new critiques was valid (we assume the intended critiques are valid).

We find that:

1. With assistance, labelers find the intended critiques $45 \pm 5\%$ of the time, whereas without assistance they only find it $27 \pm 5\%$ of the time.
2. The fraction of critiques the author considers valid does not appreciably decrease with assistance (71% with assistance versus 73% without).
3. However, the number of valid critiques is still much higher, 0.96 ± 0.09 vs. 0.62 ± 0.07.
4. With assistance, labelers also find more valid and novel critiques, 0.24 ± 0.06 vs. 0.18 ± 0.05.

3.4 Dataset release

We release a comprehensive dataset of results[4]. This includes the assistance provided, critiques used and written, ratings given, and the intended critiques. Random samples from this dataset can be found in Appendix F.2.

4 Critique quality results

In this section, we present a number of other results on critique quality. We find that critique quality is enabled by scale:

[4] https://openaipublic.blob.core.windows.net/critiques/assistance.jsonl.gz

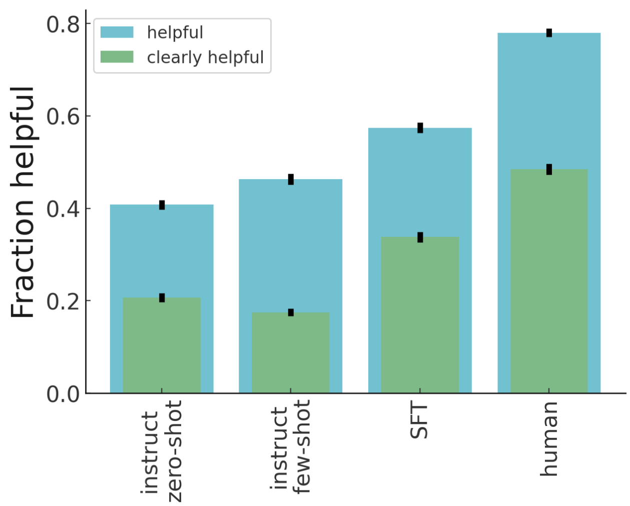

Figure 3: Our model gives more helpful critiques than InstructGPT baselines, but still significantly less helpful critiques than humans.

1. Larger models' critiques are rated as more helpful by humans. This holds even if making the answer distribution correspondingly difficult to critique by asking them to self-critique.

2. Larger models are able to improve outputs using critique-conditional refinements. We verify the critique is helping by comparing to a direct refinement baseline.

4.1 Helpfulness

The simplest way to measure critique quality is by looking at helpfulness as judged by human labelers. To check that our supervised fine-tuned model is not overly nit-picky, we also asked labelers to mark whether each critique was clearly and unambiguously helpful.

We compare our best critique model to human-written critiques, and to baseline models. For baselines, we use a model trained in the style of InstructGPT [OWJ+22] from the same pretrained model. We use this model both using a zero-shot instruction-based context, and with few-shot contexts in the style of [RWC+19, BMR+20]. For this evaluation, answers were generated randomly from either one of our large fine-tuned models, or an InstructGPT baseline model with zero-shot or few-shot prompting. We then evaluated on answers for which humans found critiques ("critiqueable answers").

Overall we find our model's critiques to be helpful more often than the baselines, but still substantially less helpful than human critiques (Figure 3). We found the InstructGPT models to give surprisingly helpful critiques, considering that they were not trained on our task at all.

4.2 Self-critiquing helpfulness and scaling

In Section 3.4, we showed that models are able to help humans find critiques on the distribution of answers coming from the same model.

One natural question to ask is: Should a model be able to reliably find flaws in its own outputs? After all, if it understands these flaws, it could have perhaps avoided them in the first place. However, there is at least one major reason you still might expect a model to identify its own mistakes: Recognizing errors is easier than avoiding them. Equivalently, verifying solutions is easier than finding them (compare to $\mathsf{P} \subseteq \mathsf{NP}$ from computational complexity theory).

It's possible that our model can identify and critique all of its mistakes. This motivates us to look at the percentage of the time poor outputs have helpful critiques. The higher this percentage, the easier it will be to assist humans in evaluation of the base task.