UNcommonsense Reasoning: Abductive Reasoning about Uncommon Situations

Wenting Zhao[1*] Justin T. Chiu[1] Jena Hwang[2] Faeze Brahman[2] Jack Hessel[2]
Sanjiban Choudhury[1] Yejin Choi[2,3] Xiang Lorraine Li[4*] Alane Suhr[5*]

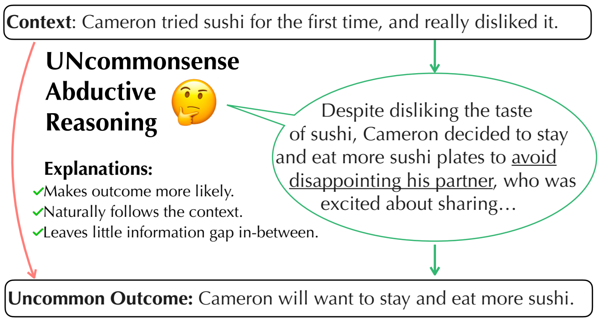
[1]Cornell University, [2]Allen Institute for Artificial Intelligence
[3]University of Washington, [4]University of Pittsburgh, [5]University of California, Berkeley
wz346@cornell.edu, xianglli@pitt.edu, suhr@berkeley.edu

Abstract

Language technologies that accurately model the dynamics of events must perform commonsense reasoning. Existing work evaluating commonsense reasoning focuses on making inferences about common, everyday situations. To instead investigate the ability to model **un**usual, **un**expected, and **un**likely situations, we explore the task of **un**commonsense abductive reasoning. Given a piece of context with an unexpected outcome, this task requires reasoning abductively to generate an explanation that makes the unexpected outcome more likely in the context. To this end, we curate and release a new English language corpus called **UNcommonsense**. We characterize the performance differences between human explainers and the best performing large language models, finding that model-enhanced human-written explanations achieve the highest quality by trading off between specificity and diversity. Finally, we experiment with several imitation learning algorithms to train open and accessible language models on this task. When compared with the vanilla supervised fine-tuning approach, these methods consistently reduce lose rates on both common and uncommonsense abductive reasoning judged by human evaluators.

1 Introduction

The ability to perform commonsense reasoning is crucial for understanding the dynamics of everyday events, both for humans and for natural language processing systems. However, most existing commonsense reasoning benchmarks focus on the ability to model common events (Sap et al., 2019; Talmor et al., 2019; Lin et al., 2020b), i.e., *given a commonly encountered situation, what commonsense inferences can be made?* Comparatively less effort has been devoted to evaluating a different class of inputs: unusual scenarios, improbable situations, and implausible events.



Context: Cameron tried sushi for the first time, and really disliked it.

Despite disliking the taste of sushi, Cameron decided to stay and eat more sushi plates to avoid disappointing his partner, who was excited about sharing…

✓Makes outcome more likely.
✓Naturally follows the context.
✓Leaves little information gap in-between.

Uncommon Outcome: Cameron will want to stay and eat more sushi.

Figure 1: Given a context and an uncommon outcome, uncommonsense abductive reasoning aims to produce an explanation so that the unlikely outcome becomes likely. The explanation needs to follow the three rules noted with the check marks.

Understanding and reasoning about these situations is crucial for the fairness and reliability of language technologies. For example, most LLMs are trained on English data. They are accustomed to Western cultural norms, and therefore non-English culture could be considered uncommon in current LLM-based NLP systems, e.g., wearing shoes indoors is normal in Western culture, but is often viewed as disrespectful in Asian households. Being able to reason about uncommon situations helps LLMs serve individuals from diverse cultural backgrounds more effectively. Uncommon situations could also be associated with important and high-risk scenarios (Weidinger et al., 2022). Consider a situation where an individual tries out a massage chair and subsequently develops small, itchy, and red welts on their back. One explanation may be that this person is allergic to vibrations, a rare yet real medical condition called vibratory urticaria. While this is an uncommon situation, an NLP system that incorrectly interprets or handles this situation could lead to severe consequences, for example a misdiagnosis of a more common condition unrelated to the chair.

To bridge this gap, we introduce UNCOMMONSENSE, a benchmark that explicitly challenges

[*] Wenting, Lorraine, and Alane's work done at AI2. Lorraine and Alane are co-last authors.

Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 8487–8505
June 16-21, 2024 ©2024 Association for Computational Linguistics

models to reason about implausible, yet still possible, events. UNCOMMONSENSE is an English-language corpus consisting of 20k unique contexts paired with explicitly uncommon outcomes. We source uncommon outcomes from the incorrect answers in several multiple choice commonsense reasoning benchmarks, which were designed to challenge models to identify the most likely outcome among multiple candidates, given a context. Given these contexts and uncommon outcomes, we crowdsource 41k abductive explanations, which provide a plausible explanation of how an uncommon outcome could have arisen, given an input context. See Figure 1 for an example. UNCOMMONSENSE complements existing commonsense reasoning corpora (e.g., Mostafazadeh et al., 2016a; Bhagavatula et al., 2020; Rudinger et al., 2020) that focus on reasoning about common events.[1]

We examine the gap between human and model performance in generating abductive uncommonsense explanations, finding subtle differences in explanation quality. Given a few demonstrations, the top-performing LLM GPT-4 (OpenAI, 2023) produces more specific explanations than those acquired through crowdsourcing; however, these explanations are less diverse. While their explanations often lack sufficient details to connect contexts to outcomes, workers recruited through crowdsourcing excel at creating a broader picture of possible intermediate events. To combine the creativity of human authors and the specificity of LLM-generated explanations, we experiment with using an LLM to refine crowd-authored explanations by filling in more details. Though LLM-generated explanations are generally preferred over the original crowd-written explanations, we find that LLM-refined crowd-written explanations hold a notable advantage over those generated only by an LLM.

Generating abductive explanations for uncommon outcomes *without* conditioning on a human-written starting point remains a challenge, particularly for publicly available models. Specifically, we find that the purely offline learning approach of supervised fine-tuned models suffer from compounding errors during generation. This is particularly problematic for our task, which generally requires lengthy explanations that bridge the gap between a context and an uncommon outcome. To this end, we experiment with two online imitation learning methods to improve the performance of open and accessible language models on abductive reasoning. When compared with supervised fine-tuning, these methods show an absolute 10% increase in win rates against the strong GPT-4 baseline when evaluated by workers on both commonsense and uncommonsense abductive reasoning.

2 Uncommonsense Abductive Reasoning

Given a natural language context x and outcome y, the task of abductive reasoning requires generating a natural language explanation z that augments the context, making the outcome more probable (Bhagavatula et al., 2020). In uncommonsense abductive reasoning, we focus on situations where an outcome y is very unlikely to happen in context x. For example, in Figure 1, our context "*Cameron tried sushi for the first time, and really disliked it.*" is paired with the unlikely outcome "*Cameron will want to stay and eat more sushi.*". One possible abductive explanation of this outcome is that "*... Cameron decided to stay and eat more sushi plates to avoid disappointing his partner, who was excited about sharing...*". When the context is augmented with this explanation, it becomes significantly more likely that the outcome will occur.

To our knowledge, no existing datasets explicitly study abductive reasoning for uncommon situations. We fill this gap by collecting the UNCOMMONSENSE dataset, which contains contexts paired with both uncommon outcomes and explanations that rationalize these uncommon outcomes. Table 1 presents several examples from UNCOMMONSENSE, with explanations written by humans. In this section, we describe our process for collecting UNCOMMONSENSE, including collecting uncommon outcomes and abductive explanations.

2.1 Uncommon Outcomes

We first collect pairs of contexts and uncommon outcomes. We source contexts from two existing commonsense datasets: SocialIQA (Sap et al., 2019) and ROCStories (Mostafazadeh et al., 2016b). Each uncommon outcome is either human-written or LLM-generated.

un-SocialIQA. SocialIQA is a multiple-choice question answering dataset created to evaluate reasoning about social interactions. Each example consists of a context x, a question q, and three answer choices $\mathcal{A}$, one of which is correct. To pick the uncommon outcome, we identify the least likely answer choice (among the incorrect ones) by

[1] Data is available at huggingface.co/datasets/allenai/UNcommonsense

Context	Uncommon Outcome	Explanation
Kai bought a Kindle from Amazon and used it all of the time.	Kai will want to return the Kindle and go back to reading physical books only.	After a month of reading books with the Kindle, the free book trial ran out and Kai decided that reading books on the Kindle was not worth paying for. The return period for the Kindle has not ended yet.
Tracy went shopping at the market and brought many good items at the super market like fish and meat.	Tracy will want to get angry.	Tracy realized that many of the items she bought were already expired, and the shopkeepers had knowingly sold her expired meats.
Scott was hungry. He decided to cook dinner. He cooked tacos. He made enough to share with a friend.	His friend was so offended he asked Scott to leave.	Scott made the tacos with beef and didn't tell his friend until after they ate, even though he know that his friend was a strict vegetarian.
Drew order a small ice cream cone at the Drive Thru. He drove to the front window to wait for his order. The cashier handed the ice cream with one of her hands. Drew took the ice cream cone and turned it upside down.	Drew kept his car clean this way.	He just dumped the whole thing into a small plastic cup he kept in the car and then he ate it out of the cup.

Table 1: UNCOMMONSENSE examples. The first two examples are from **un**-SocialIQA and the next two examples come from **un**-RocStories; explanations are written by crowdworkers.

we computing $\texttt{argmin}_{a \in \mathcal{A}^-} p(a|x, q)$ with GPT-3, where $\mathcal{A}^-$ is the set of two incorrect answers. We then use LLM prompting[2] to combine the question and the least likely *incorrect* answer choice into a declarative sentence, which we take as the uncommon outcome y.

All original SocialIQA answer choices are human-written. To further diversify uncommon outcomes, we additionally generate new improbable answer choices using few-shot prompting with LLMs. We use 6-shot prompting with GPT-4[3] to produce one improbable answer for a randomly sampled subset of SocialIQA contexts and questions, then combine the question and generated answer into into uncommon outcomes using the same procedure above.

un-RocStories. The ROCStories Cloze Test includes examples of four-sentence stories paired with two sentence-length endings. The original task is to predict which of the two endings is more likely. In UNCOMMONSENSE, we take each four-sentence story as the context x and the *incorrect* ending as the uncommon outcome y.

Filtering out common outcomes. To focus on uncommon scenarios, we exclude examples where outcomes are obvious in the context.[4] We prompt GPT-4 to rate the likelihood of the outcome given the context on a scale from 1 to 5, and remove examples with ratings of 4 or 5. Filtering with this criterion removes 0.7% of **un**-RocStories examples and 1.82% of **un**-SocialIQA examples.

2.2 Explanations for Uncommon Outcomes

We crowdsource explanations of uncommon outcomes z on Amazon Mechanical Turk (MTurk) from 156 unique workers, with a pay rate of 15 USD/hour.[5] We also experiment with using an LLM both to generate explanations from scratch given contexts paired with uncommon outcomes, and to enhance crowd-written explanations. Specifically, we use GPT-4, which has demonstrated strong reasoning abilities on a wide range of tasks.

Explanation Writing. We first conduct a paid qualification task that identifies 204 workers who write high-quality explanations, who are then invited to participate in explanation writing tasks. Tasks are launched in small batches, and we evenly distribute tasks across workers in each batch, which, by design, ensures that no worker writes too many explanations. Due to the subjectivity on evaluation for this task, we emphasize collecting a wide variety of explanations on the development and test sets, creating no less than three tasks for each pair

[2] All prompting templates can be found in Appendix D.

[3] We use gpt4-0314 for all generation tasks, including uncommon outcomes, explanations, and during online learning.

[4] Both human-written and LLM-generated outcomes can be too obvious without filtering.

[5] Appendix E contains additional details on crowdsourcing.

	un-RocStories	**un**-SocialIQA
# of context-outcome (x, y) pairs, with y sourced from...		
Human	1,775 / 765 / 999	5,531 / 543 / 999
LLM	0 / 0 / 0	8,699 / 931 / 705
# of explanations z, sourced from...		
Crowd	8,428 / 4,240 / 4,835	14,563 / 4,407/ 5,238
C+LLM	8,333 / 4,203 / 4,771	14,469 / 4,390 / 5,209
LLM	17,548 / 7,556 / 9,919	14,324 / 4,422 / 5,112

Table 2: Basic statistics of UNCOMMONSENSE. Counts in cells report the number of examples split across the train/dev/test sets.

of context and outcome collected in Section 2.1. We also perform extensive quality control on collected explanations, described in Appendix E. We also use this task to identify the outcomes that are impossible given their contexts, asking workers to mark these examples and provide their reasoning. We remove examples marked as impossible by more than half of its annotators.

LLM-Enhanced Crowd-written Explanations. We prompt LLMs to enhance crowd-written explanations. We instruct GPT-4 to add details that better connect contexts and outcomes.

LLM-Generated Explanations. We use 3-shot prompting with GPT-4 to generate explanations for each context-outcome pair.

LLM-Enhanced LLM-Generated Explanations. To directly investigate the effect of LLM-based explanation enhancement, we also apply LLM enhancement to one randomly-chosen *LLM* explanation for each context-outcome pair, using the same prompting method that was used to enhance *Crowd* explanations. We refer to these LLM-enhanced LLM-generated explanations as *LLM*^{2}.

3 Data Analysis

Table 2 contains basic statistics of the collected data. UNCOMMONSENSE includes 3,539 contexts paired with uncommon outcomes in **un**-RocStories and 17,408 in **un**-SocialIQA for a total of 20,947 context-outcome pairs. We adopt the same train/dev/test splits as the original releases of RocStories and SocialIQA. In total, we collect 41,711 crowd-written explanations (*Crowd*), 41,375 LLM-enhanced crowd-written explanations (*C+LLM*), and 58,881 LLM-generated explanations (*LLM*). We compare explanations from these three sources using several metrics, including human preference judgments, explanation lengths, and measures of explanation diversity.

	UNCOMMONSENSE			αNLG
l	**un**-RocStories	**un**-SocialIQA		
		(Human)	(LLM)	
5	0.0	0.0	0.0	0.1
4	0.0	0.0	0.0	31.8
3	29.4	50.7	25.8	40.3
2	63.1	42.1	59.6	19.9
1	7.5	6.9	14.5	0.9

Table 3: Proportion of outcomes assigned likelihoods $l \in \{1 \dots 5\}$ for examples in UNCOMMONSENSE corresponding to **un**-RocStories and **un**-SocialIQA (split by human-authored and LLM-generated uncommon outcomes), compared with αNLG.

Unlikely Outcomes. We utilize GPT-4 prompting to quantify, on a scale from 1 to 5, how likely an outcome may occur given the context. Table 3 summarizes the ratios of outcomes broken down by their scales with 1 being the most unlikely. In αNLG, only 20.8% of outcomes have a scale of 1 or 2. Significantly more outcomes are rated 1 or 2 in **un**-RocStories (70.6% of outcomes) and **un**-SocialIQA (49.0% of human-written and 74.1% of LLM-generated outcomes). Compared to αNLG, UNCOMMONSENSE poses a unique challenge of abductive reasoning about uncommon outcomes.

Explanation Preferences. We first compare pairwise preferences of *LLM* explanations versus *Crowd*, *C+LLM*, and *LLM*^{2} explanations. We randomly sample 500 context-outcome pairs from each UNCOMMONSENSE test set, and select the same explanation from *LLM* that was randomly chosen to be enhanced into *LLM*^{2}. We then randomly sample a single crowd-written explanation for each pair from *Crowd*, along with its enhanced counterpart in *C+LLM*. This selection procedure allows us to directly compare the effect of applying LLM-based enhancement to both crowd-written and LLM-generated explanations.

We recruit crowdworkers who provided quality explanations during data collection to provide pairwise preferences between *Crowd*, *C+LLM*, and *LLM*^{2} explanations with *LLM* explanations based on the same rules used for the explanation-writing task (Section 2.2).[6] Raters can select one of the two explanations as better, or can mark ties between the two as equally bad or equally good. Table 4 shows that *Crowd* explanations are least often preferred and *C+LLM* explanations are the most preferred. While *LLM* can improve via LLM-based enhance-

[6] Figure 12 in the appendix shows the MTurk preference evaluation template.

	un-SocialIQA			**un**-RocStories		
	Crowd	*C+LLM*	*LLM*^{2}	*Crowd*	*C+LLM*	*LLM*^{2}
Win	30.8	43.2	33.8	19.2	28.4	26.4
Eql. good	33.4	34.8	41.2	37.0	45.6	42.4
Eql. bad	3.4	2.0	3.8	12.0	3.0	3.0
Lose	32.4	20.0	21.2	42.6	23.0	28.2
Fleiss' κ	0.47	0.47	0.55	0.48	0.43	0.53

Table 4: Preference judgments given by crowdworkers comparing explanations from *LLM* with explanations from *Crowd*, *C+LLM*, and *LLM*^{2}.

l	**un**-SocialIQA	**un**-RocStories
2	77.35	71.63
1	90.90	75.85

Table 5: Non-lose rates of *C+LLM* versus *LLM*, broken down by the likelihoods l of outcomes ($l = 2$ is more likely, and $l = 1$ is less likely).

Context: The band walked to the front of the stage. They began to perform. The electricity immediately went off. Everyone couldn't see where they were.

Outcome: People danced in the well lit room.

GPT4-generated Explanation (*LLM*): Someone in the audience had a powerful flashlight, and they used it to illuminate the room while the band continued to play acoustically. This allowed everyone to continue dancing despite the power outage. *Comment: The explanation is detailed, but it is a less likely continuation of the context.*

Crowd-written Explanation (*Crowd*): Suddenly, they found the light switch. *Comment: The explanation is likely to happen, but there is a large information gap in-between.*

Crowd-GPT4 Explanation (*C+LLM*): The band's manager quickly grabbed a flashlight and located the circuit breaker, restoring power to the venue. With the electricity back on, the lights illuminated the room, allowing everyone to see and continue dancing to the band's performance. *Comment: Starting with the crowd-written explanation and refining it with an LLM results in plausible explanations that include sufficient details to connect the context and outcome.*

Figure 2: Qualitative comparison between *LLM* explanations, *Crowd* explanations, and *C+LLM* explanations. In *Comments*, we make connections to the three rules in explanation writing.

	un-SocialIQA	**un**-RocStories
Crowd+LLM	44	35
Eql. good	33	30
Eql. bad	2	1
LLM	21	34

Table 6: Comparing *Crowd+LLM* explanations to *LLM* explanations when both LLMs and crowdworkers are provided the same instrutions for producing explanations.

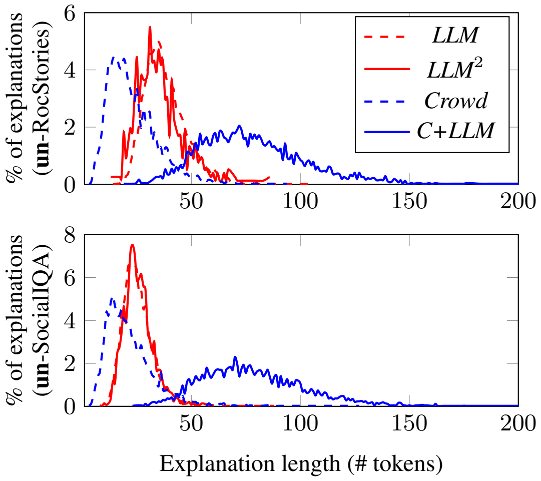


Figure 3: Distribution of explanation lengths in **un**-RocStories (top) and **un**-SocialIQA (bottom), computed on the development sets of each data subset.

ment, these explanations are still less preferred when compared to *C+LLM*. Finally, we include the Fleiss' κ score to demonstrate the inter-annotator agreement rate between workers, where they all fall within the range from 0.40 to 0.60.[7] Figure 2 compares explanations generated by *LLM*, *Crowd*, and *C+LLM* for an example in **un**-RocStories. Table 5 presents the non-lose rates of *C+LLM* explanations against *LLM* explanations broken down by likelihoods.[8] *C+LLM* explanations are preferable as the likelihood of outcomes are less likely.

We note that in the analysis above, we provide LLMs and crowdworkers with different instructions for producing explanations. The instructions given to crowdworkers are more detailed than those given to the LLMs. We further explore if giving LLMs the same instructions we give to humans will make LLMs perform better. We compare *Crowd+LLM* and *LLM* explanations and present the results in Table 6. We find that this instruction improves explanations on **un**-RocStories but harms explanations on **un**-SocialIQA. Therefore, LLMs still cannot always benefit from detailed instructions even when they include more information on what are considered good explanations.

Quantitative Comparison of Explanations. We investigate several distributional differences across the four sources of explanations. Figure 3 shows

[7] Our preference-based ranking is a four-way classification. Even though scores between 0.40 and 0.60 are considered moderate agreement for the two-class case, it is more challenging to achieve these scores in the four-class case.

[8] The 100 test examples considered here only contain a significant number of outcomes with likelihoods $l = 1, 2$.

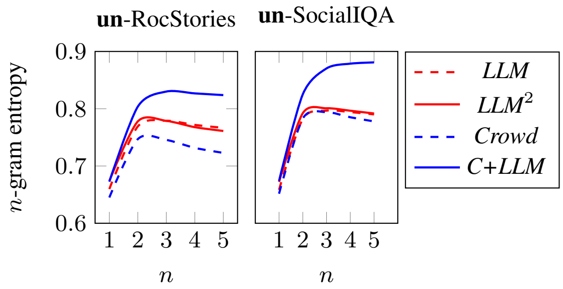


Figure 4: Entropies of n-gram distributions in **un**-RocStories (left) and **un**-SocialIQA (right), computed on the development sets of each data subset.

the distribution of explanation lengths.[9] *Crowd* explanations are significantly shorter than *LLM*, with an average length of 22.9 ± 11.3 tokens per explanation in **un**-RocStories and 22.0 ± 11.9 in **un**-SocialIQA, compared to an average of 38.2 ± 9.9 and 25.5 ± 7.1 respectively for *LLM*. However, enhancing crowd-written explanations with an LLM significantly increases their lengths over *LLM*: *C+LLM* has an average explanation length of 78.0 ± 24.4 tokens in **un**-RocStories and 78.3 ± 23.5 in **un**-SocialIQA. This pattern does not hold for LLM-based enhancement of LLM-generated explanations: *LLM*^{2} has average lengths of 35.6 ± 10.8 and 25.9 ± 6.7 respectively, not significantly different from *LLM*. Therefore, length of the explanations produced by *C+LLM* can vary significantly.

In Figure 4, we investigate the entropy of the distribution of n-grams from $n \in \{1, \dots, 5\}$ across the different sources of explanations.[10] We use entropy as a measure of lexical diversity (Jung et al., 2023). We find trends similar to the analysis of explanation lengths: while *Crowd* has generally lower entropy than *LLM*, LLM enhancement of crowd-written explanations results in significantly higher entropy (*C+LLM*), while it has no effect on LLM-generated explanations (*LLM*^{2}). Therefore, *C+LLM* results in the highest lexical diversity in explanation writing.

Finally, in addition to using n-grams as a measure of diversity, we also perform embedding analysis to evaluate the semantic diversity of explanations written by crowdworkers and GPT-4. In particular, we compute the embedding of each *crowd* explanation and each *LLM* explanation[11], and we compute the distance between every pair of explanations for *crowd* explanations and *LLM* explanations, respectively. We find that the average distance between *LLM* explanations is 1.26 ± 0.058, while the average distance between *crowd* explanations is 1.29 ± 0.052, suggesting that *crowd* explanations are more semantically diverse than *LLM* explanations.

4 Imitation Learning for Abductive Reasoning

Existing methods for abductive reasoning focus on performing supervised fine-tuning (SFT) with a static dataset (Bhagavatula et al., 2020; Rudinger et al., 2020). Training using static demonstration data is vulnerable to exposure bias: during training, the model learns to predict the next token in an explanation conditioned on a gold-standard prefix; however, when the model generates an entirely new explanation during inference, it is conditioned on its own previously generated tokens. This inconsistency between training and inference procedures leads to error propagation at inference time, and a reduction in the quality of explanations. To address this issue, we experiment with several on-policy imitation learning algorithms.

4.1 Background: Imitation Learning

In the task of abductive reasoning, a policy π maps from the context x, an outcome y, and the prefix sequence of an explanation z to a distribution over the output vocabulary. Explanations are generated token-by-token, with the jth token $z_j \sim \pi(\cdot \mid x, y, z_{:j-1})$, and the entire explanation sampled from π as $z \sim \pi(\cdot \mid x, y)$.

Let $\pi^*(\cdot)$ be an expert policy and $\pi_\theta(\cdot)$ be a learner policy with parameters θ. The objective of imitation learning is to find parameters θ that result in the learner policy assigning high probabilities to expert demonstrated explanations.

Behavior Cloning (BC). BC uses expert demonstrations $\mathcal{D} = \{(x, y, z)\}^N$ and a supervised learning objective that train a learner policy to maximize the probability of expert demonstrations. Existing methods of using SFT is a type of behavior cloning. A drawback of BC is the aforementioned exposure bias problem; as a result, errors are more likely to propagate during inference, where the learner fails

[9] We use `nltk.wordpunct_tokenize` (Bird et al., 2009) for tokenizing explanations.

[10] As different data sources contain a different number of explanations per context-outcome pair, we compute entropy using 1,000 iterations of bootstrap sampling of one explanation per context-outcome pair in each data subset.

[11] We compute the embeddings using the OpenAI ada embedding model (`text-embedding-3-large`).

Algorithm 1 EaO: Using expert as an oracle.

```
Inputs: Initial learner policy parameters $\theta_0$, expert policy $\pi^*(\cdot)$, dataset $\mathcal{D} = \{(x, y)\}^N$, block size $k$, initial prefix size $b$, number of training epochs $I$.
$\tilde{\mathcal{D}} \leftarrow \emptyset$
for $i = 0, \ldots, I - 1$ do
    for $(x, y) \in \mathcal{D}$ do
        $\tilde{z} \sim \pi_{\theta_i}(\cdot \mid x, y)$
        $z^* \sim \pi^*(\cdot \mid x, y, \tilde{z}_{:b})$
        $\tilde{\mathcal{D}} \leftarrow \tilde{\mathcal{D}} \cup \{(x, y, \tilde{z}_{:b} z^*)\}$
    end for
    $\theta_{i+1} \leftarrow \theta_i$ further optimized on $\tilde{\mathcal{D}}$ with supervised learning.
    $b \leftarrow b + k$
end for
Returns: Learned policy parameters $\theta_I$.
```

to recover from its own mistakes, as it was never exposed to these mistakes during training.

Online Learning. To address the exposure bias problem for sequence prediction tasks, Ross et al. (2011) propose DAgger, where an expert policy is used at training time to provide oracle continuations to learner-generated prefixes. The learner policy is then optimized to maximize the probability of oracle continuations, conditioned on sequence prefixes generated by the learner. DAgger and its variants have been used in many NLP tasks, including dependency parsing (Goldberg and Nivre, 2012), instruction following (Anderson et al., 2017), and language generation (Lin et al., 2020a).

4.2 Imitation Learning for Abductive Reasoning

We explore two online imitation learning approaches that assume different levels of access to an expert policy, which is in our case a top-performing LLM. First, we assume access to the expert policy at any point during training, which allows us to use it as an oracle. Next, we consider a setting where the expert may not be available at training time (e.g., for cost reasons), and we only have a static set of expert demonstrations.

EaO: Using expert as an oracle on-line. Algorithm 1 formalizes our DAgger-inspired algorithm, which we call "Expert as Oracle" (EaO). We train with I total epochs over the training dataset $\mathcal{D} = (x, y)^N$. Throughout learning, we maintain a training dataset $\tilde{\mathcal{D}}$ containing examples of contexts and outcomes paired with explanations aggregated during each epoch. In each epoch i, and for each example (x, y), we use the current learner parameters θ_i to sample an explanation $\tilde{z}$. Using a prefix b of a fixed size, we then sample a continuation of $\tilde{z}_{:b}$ using the expert policy π^*. Finally, we add an example to $\tilde{\mathcal{D}}$ that concatenates the first b tokens of the learner's sample with the expert's completion. After aggregating examples for the epoch, we apply supervised training on examples in $\tilde{\mathcal{D}}$ to acquire updated parameters θ_{i+1}. After each epoch, we increase the length of the prefix generated by learner policy b by a fixed block size k.

SED: Using only static expert demonstrations. For the setting where we have access only to a static set of expert demonstrations, we propose an online learning algorithm that similarly aims to avoid the exposure bias problem of behavior cloning.[12]

We modify the loss function of behavior cloning, which maximizes the probability of expert demonstration z, by adding two terms: (a) a term that minimizes the probability of explanations generated by the learner policy during training $\tilde{z}$; and (b) the KL divergence from initial policy for stabiling the training process (Schulman et al., 2017). Formally, after sampling $\tilde{z}$ for each instance at each iteration from the current policy, we optimize:

$$\mathcal{L}(\theta) = \frac{1}{N} \sum_{(x,y,z,\tilde{z}) \in \tilde{\mathcal{D}}} \Big\{ -\log \pi_\theta(z|x, y) + \lambda \log \pi_\theta(\tilde{z}|x, y) + \beta \mathrm{KL}\left(\pi_{\theta_0}(\cdot|x, y, z_{<t}) || \pi_{\theta_i}(\cdot|x, y, z_{<t})\right) \Big\} \quad (1)$$

5 Experiments

Evaluation. We evaluate the proposed imitation learning methods with three sets of metrics. We focus on preference-based pairwise evaluation judged by humans.[13] We report performance on the same 100 randomly-sampled examples.[14] In Appendix C, we report two additional sets of metrics: (a) human judgements on seven binary questions (e.g., is the outcome more likely given the context and the explanation than given the context alone?) that evaluate different failure modes, and (b) a number of reference-based automatic evaluation metrics, e.g. BERTScore (Zhang et al., 2020b).

Base models. As baselines, we experiment with 3-shot prompting with GPT-3 (Brown et al., 2020) and, following the state-of-the-art approach for commonsense abductive reasoning (Khashabi

[12] Full pseudocode is in Appendix G.

[13] For simplicity, in this evaluation, we report equally good and equally bad as the same category (Tie).

[14] We will maintain a leaderboard that provides human evaluation of these examples on model-generated explanations for two years. We will also maintain the same human annotator pool to increase reproducibility and ensure fairness.

Supervision	Base Model	**un**-RocStories Win	 Tie	 Lose	**un**-SocialIQA Win	 Tie	 Lose
3-shot prompting	GPT-3	13	20	67	33	13	54
SFT with *LLM*	GPT-2-XL	6	22	72	7	44	49
	LLaMA-7B	13	35	52	25	38	37
	FlanT5-XXL	16	28	56	16	47	37
SFT with *C+LLM*	GPT-2-XL	6	26	64	13	32	55
	LLaMA-7B	21	31	48	19	39	42
	FlanT5-XXL	11	32	57	27	32	41

Table 7: Experimental comparison of GPT-3 using few-shot prompting, and *SFT* with two sources of training explanations on three different base models , using pairwise preference-based evaluation on the test set of *LLM*.

Task	Uncommon?	Sources of Explanations	# of Explanations
αNLG	N	Crowd workers	76k
d-NLI	N	Crowd workers	200k
Arnaout et al.	N	Variants of BERT models	N/A
TODAY	Y	Crowd workers	2.2k
Collins et al.	Y	Crowd workers	0.8k
UNCOMMONSENSE	Y	Crowd workers and GPT-4	41k

Table 8: Summary of the differences between the proposed dataset and the existing datasets.

		Win	Tie	Lose
un-RocStories	*SFT*	6	22	72
	SED	12	24	64
	EaO	17	16	67
un-SocialIQA	*SFT*	7	44	49
	SED	9	34	57
	EaO	13	39	48
αNLG	*SFT*	13	20	67
	SED	14	23	63
	EaO	14	23	63
Sen-Making	*SFT*	12	41	47
	SED	13	49	38
	EaO	13	52	35

Table 9: Comparison between different imitation learning methods using pairwise preference-based evaluation on the test set of *LLM*.

et al., 2022), on several open and accessible language models: FlanT5-XXL (Chung et al., 2022), LLaMA-7B (Touvron et al., 2023), and GPT-2-XL (Radford et al., 2019). To compare the benefit of different sources of training data, we perform *SFT* on explanations in the training sets of *LLM* (LLM-generated explanations) and *C+LLM* (LLM-enhanced crowd-written explanations). Because *Crowd* (crowd-written explanations) are the least preferred subset in UNCOMMONSENSE, we do not fine-tune on them. Appendix F contains additional experimental details.

Can imitation learning improve a given model? We apply our proposed imitation learning algorithms, EaO and SED, to GPT-2-XL as the initial learner policy. This is the weakest (but most computationally accessible) base language model of the three we consider for *SFT*. This choice is purposeful, as our experiment intends to assess whether imitation learning can improve a *given* LM. For a fair comparison, we use the same expert policy (GPT-4) for both *EaO* and *SED*. In addition to uncommonsense benchmarks, we report performance on two commonsense benchmarks, αNLG (Bhagavatula et al., 2020) and Sense-making (Wang et al., 2019) to show generalization of the methods.

5.1 Results

Baselines. Table 7 shows the performance of the baseline systems. Unsurprisingly, explanations generated from few-shot GPT-3 are rarely preferred by crowdworkers to those GPT-4 itself generated (13% of the time). However, GPT-3 also underperforms the 25x smaller (but supervised fine-tuned) LLaMA-7B (48% non-lose rate vs. GPT-4) and 16x smaller FlanT5-XLL (44% non-lose rate vs. GPT-4). In addition, having *C+LLM* to be supervision sometimes leads to better performance than using *LLM* as supervision but in other times hurts. We hypothesize that despite *LLM* explanations being worse than *C+LLM* explanations, they are easier for the small models to learn. Finally, all methods but one still lose to *LLM* explanations, indicating that *SFT* alone is insufficient.

Imitation Learning. Table 9 shows the performance comparing *SFT* with the two imitation learning methods, *SED* and *EaO*, on four datasets when using GPT-2-XL as the base moddel. On both UNCOMMONSENSE and commonsense benchmarks, *SED* and *EaO* show strong improvements against *SFT* by reducing the losing rate to *LLM* explanations or by increasing the win rates. Except for αNLG, *EaO*, which trains using expert online corrections to learner-generated sequence prefixes, shows more promise than *SED* on most of the datasets. However, *SED*, which is no more costly than *SFT*, can significantly improve the performance of the weak-but-accessible base model GPT-2-XL on both commonsense and uncommonsense reasoning except on **un**-SocialIQA.

6 Related Work

αNLG (Bhagavatula et al., 2020) is the most closely related task to UNCOMMONSENSE: both require generating explanations to bridge contexts and outcomes (except αNLG focuses on common, everyday scenarios). d-NLI (Rudinger et al., 2020) consider a related task of generating an explanation explanation that weakens an outcome. Additional works cover methods for generating explanations, e.g., Du et al. (2022), Zhou et al. (2021), Wang et al. (2019), Zhang et al. (2020a), inter alia.

Reasoning about uncommon but possible scenarios has been studied in other settings. Arnaout et al. (2022) propose a method for identifying informative negations about everyday concepts in large-scale commonsense knowledge bases. Tang et al. (2023) present a decoding method for producing less plausible explanations for everyday events. Collins et al. (2022) create a small-scale benchmark containing about 800 curated uncommon statements, along with explanations making sense of these statements. UNCOMMONSENSE differs in structure and focus from these prior works. Finally, TODAY (Feng et al., 2023) proposes a temporal reasoning task to study the order of two events. Atypical order of two events could be uncommon, and justifying the order is uncommonsense reasoning. Because UNCOMMONSENSE is not built from reversing the order of temporal events, it encompasses a different set of uncommon situations, including social reasoning, cultural reasoning, and physical reasoning. With each situation, UNCOMMONSENSE also contains more than one explanation, collected from both crowd workers and GPT-4. We summarize the differences between UNCOMMONSENSE and existing datasets in Figure 8.

Finally, uncommonsense reasoning is closely related to defeasible reasoning (Rudinger et al., 2020; Madaan et al., 2021a,b). Both defeasible reasoning and reasoning about uncommon situations are, given context x and outcome y, finding an explanation z that changes the original likelihood $p(y|x)$ by adding z: $p(y|x, z)$. However, we note that feasible reasoning itself does not place any constraint on $p(y|x)$. Reasoning about uncommon situations falls on the long-tail distribution of defeasible reasoning as it focuses on the cases where $p(y|x)$ is very small.

7 Conclusion

We propose a new task, uncommonsense abductive reasoning, designed to assess the ability of NLP systems to reason about uncommon scenarios in abductive reasoning tasks. We explore two imitation learning methods to improve accessible language models on uncommonsense abductive reasoning. Experiments show that access to expert behavior, particularly when using the expert as an oracle in online training, significantly improves the explanation quality of smaller models.

Limitations

While our dataset offers advantages over existing sources, we acknowledge the following limitations. First, our dataset may suffer from social biases in the data collection process, and the labeling process may contain errors and inconsistencies. Despite best efforts to ensure high-quality annotations, occasional human errors are possible. Additionally, our dataset only contains uncommon situations in English and thus lack of language diversity. Finally, our main preference-based evaluation relies on human evaluators, which can be less producible and costly. There is thus a large room for improvement for more effective and affordable evaluation methods.

Ethics Statement

This work aims to advance NLP and commonsense reasoning by introducing a new benchmark, UNCOMMONSENSE, which investigates abductive reasoning about uncommon events. It is important to study these uncommon situations as they provide valuable insights into the proper functioning of AI systems in real-world, unpredictable circumstances. However, we emphasize the need to ensure that the generation of natural language explanations follows ethical guidelines and respects privacy, diversity, and fairness. We are committed to maintaining transparency and sharing the code and data, fostering open collaboration to address potential ethical concerns and promote the responsible advancement of AI technologies.

Acknowledgments

We thank the anonymous reviewers for their feedback to help us improve the paper. We thank Xiang Ren, Xinyan Yu, and Sasha Rush for numerous helpful discussions. This work was partially supported by an AI2 Young Investigator Grant.

References

Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian D. Reid, Stephen Gould, and Anton van den Hengel. 2017. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3674–3683.

Hiba Arnaout, Simon Razniewski, Gerhard Weikum, and Jeff Z Pan. 2022. UnCommonSense: Informative negative knowledge about everyday concepts. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 37–46.

Chandra Bhagavatula, Ronan Le Bras, Chaitanya Malaviya, Keisuke Sakaguchi, Ari Holtzman, Hannah Rashkin, Doug Downey, Wen tau Yih, and Yejin Choi. 2020. Abductive commonsense reasoning. In *International Conference on Learning Representations*.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Katherine M Collins, Catherine Wong, Jiahai Feng, Megan Wei, and Josh Tenenbaum. 2022. Structured, flexible, and robust: benchmarking and improving large language models towards more human-like behavior in out-of-distribution reasoning tasks. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 44.

Li Du, Xiao Ding, Kai Xiong, Ting Liu, and Bing Qin. 2022. e-CARE: a new dataset for exploring explainable causal reasoning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 432–446, Dublin, Ireland. Association for Computational Linguistics.

Yu Feng, Ben Zhou, Haoyu Wang, Helen Jin, and Dan Roth. 2023. Generic temporal reasoning with differential analysis and explanation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12013–12029, Toronto, Canada. Association for Computational Linguistics.

Yoav Goldberg and Joakim Nivre. 2012. A dynamic oracle for arc-eager dependency parsing. In *Proceedings of COLING 2012*, pages 959–976, Mumbai, India. The COLING 2012 Organizing Committee.

Jaehun Jung, Peter West, Liwei Jiang, Faeze Brahman, Ximing Lu, Jillian Fisher, Taylor Sorensen, and Yejin Choi. 2023. Impossible distillation: from low-quality model to high-quality dataset and model for summarization and paraphrasing.

Daniel Khashabi, Gabriel Stanovsky, Jonathan Bragg, Nicholas Lourie, Jungo Kasai, Yejin Choi, Noah A. Smith, and Daniel Weld. 2022. GENIE: Toward reproducible and standardized human evaluation for text generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11444–11458, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Alexander Lin, Jeremy Wohlwend, Howard Chen, and Tao Lei. 2020a. Autoregressive knowledge distillation through imitation learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6121–6133, Online. Association for Computational Linguistics.

Bill Yuchen Lin, Wangchunshu Zhou, Ming Shen, Pei Zhou, Chandra Bhagavatula, Yejin Choi, and Xiang Ren. 2020b. CommonGen: A constrained text generation challenge for generative commonsense reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1823–1840, Online. Association for Computational Linguistics.

Aman Madaan, Dheeraj Rajagopal, Niket Tandon, Yiming Yang, and Eduard Hovy. 2021a. Could you give me a hint ? generating inference graphs for defeasible reasoning. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 5138–5147, Online. Association for Computational Linguistics.

Aman Madaan, Niket Tandon, Dheeraj Rajagopal, Peter Clark, Yiming Yang, and Eduard Hovy. 2021b. Think about it! improving defeasible reasoning by first modeling the question scenario. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6291–6310, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016a. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 839–849, San Diego, California. Association for Computational Linguistics.

Nasrin Mostafazadeh, Alyson Grealish, Nathanael Chambers, James Allen, and Lucy Vanderwende. 2016b. CaTeRS: Causal and temporal relation scheme for semantic annotation of event structures. In *Proceedings of the Fourth Workshop on Events*, pages 51–61, San Diego, California. Association for Computational Linguistics.

OpenAI. 2023. GPT-4 technical report.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. 2011. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635. JMLR Workshop and Conference Proceedings.

Rachel Rudinger, Vered Shwartz, Jena D. Hwang, Chandra Bhagavatula, Maxwell Forbes, Ronan Le Bras, Noah A. Smith, and Yejin Choi. 2020. Thinking like a skeptic: Defeasible inference in natural language. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4661–4675, Online. Association for Computational Linguistics.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. 2019. Social IQa: Commonsense reasoning about social interactions. 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 4463–4473, Hong Kong, China. Association for Computational Linguistics.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347.

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.

Liyan Tang, Yifan Peng, Yanshan Wang, Ying Ding, Greg Durrett, and Justin Rousseau. 2023. Less likely brainstorming: Using language models to generate alternative hypotheses. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 12532–12555, Toronto, Canada. Association for Computational Linguistics.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Cunxiang Wang, Shuailong Liang, Yue Zhang, Xiaonan Li, and Tian Gao. 2019. Does it make sense? and why? a pilot study for sense making and explanation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4020–4026, Florence, Italy. Association for Computational Linguistics.

Laura Weidinger, Jonathan Uesato, Maribeth Rauh, Conor Griffin, Po-Sen Huang, John Mellor, Amelia Glaese, Myra Cheng, Borja Balle, Atoosa Kasirzadeh, et al. 2022. Taxonomy of risks posed by language models. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 214–229.

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.

Hongming Zhang, Xinran Zhao, and Yangqiu Song. 2020a. WinoWhy: A deep diagnosis of essential commonsense knowledge for answering Winograd schema challenge. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5736–5745, Online. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020b. BERTScore: Evaluating text generation with BERT. In *International Conference on Learning Representations*.

Pei Zhou, Pegah Jandaghi, Hyundong Cho, Bill Yuchen Lin, Jay Pujara, and Xiang Ren. 2021. Probing commonsense explanation in dialogue response generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4132–4146, Punta Cana, Dominican Republic. Association for Computational Linguistics.

l	Context	Outcome
4	Kate and Greg went to a little candy shop together. They looked around at their options and made their choice. They went up to the cashier and said what they wanted. The cashier, with unwashed hands, bagged the candy without gloves.	Kate and Greg licked the candy gleefully.
3	I went to the post office yesterday. It took a while to get there since it's on the other side of town. Once I got there I mailed my letters and headed home. It's always easier to get home than to get somewhere.	I could not find my way back from the post office.
2	My niece just got engaged. She is Chinese and her fiance is Caucasian. Her parents had them over for a home cooked meal. The fiance got nausea from the unfamiliar dishes and had to leave.	My niece was thrilled that her fiance was sick.
1	Josh woke up early to get ready for the hike he had been planning. After a shower, he made sure all his supplies were packed. He left his house and drove to the park where he was going hiking. Because it was early in the day Josh had the trail mostly to himself.	Josh loathed the outdoors.
4	Jordan finished their test so fast and still got an A plus as always.	Other students will be jealous.
3	Skylar gave Robin the permission to eat cake after Robin caused some trouble.	Robin will want to refuse to eat the cake.
2	Austin brought tears to Tracy's eyes when he brought her flowers.	Austin will be hated.
1	Carson threw beer in Kendall's face during a heated argument with her.	Carson will receive a medal for their behavior.

Table 10: Example outcomes of different likelihood scores $l \in \{4, 3, 2, 1\}$.

A Qualitative Analysis of Outcomes

Table 10 presents example outcomes of different likelihood scores.

B Processing Outcomes in SocialIQA

We use three types of questions: what will X want to do next, what will happen to X, and how would you describe X. We do the following steps to construct the outcome.

1. We remove the correct answer choice, and we are left with two incorrect answer choices.
2. We feed GPT3 (text-davinci-03) "{context} {question} {answer}" and compute the answer probability p (answer | context, question) and choose the answer that has the lower probability.
3. We prompt ChatGPT to combine the question and the answer to be the outcome, in the six-shot setting. When we receive a response from ChatGPT, we check whether the original answer is in the output, if it doesn't contain the answer, we send the same prompt to GPT-4. If GPT-4 still fails, we mark the example and manually combine the question and the answer. Refer to 5 for the combining prompting template.

Because SocialIQA contains many invalid answer choices, the combining step often fails (e.g., the question is "what will person X do next", and the answer is "sad"), we rely on ChatGPT to detect such cases. We throw out the examples when ChatGPT refuses to do the combination.

C More Evaluation

We include additional automatic and human evaluation results on baseline models and our proposed imitation learning methods, *SED* and *EaO*. The additional human evaluation is is a set of seven human evaluation questions that target different failure modes of generated explanations:

1. Is the explanation relevant to the context? (denoted as relevance x)
2. Is the explanation relevant to the outcome? (denoted as relevance y)
3. Is the explanation not self-contradictory? (denoted as consistency z)
4. Is the explanation not contradictory to the context? (denoted as consistency x)

Supervision	Model	Consistency x	 y	 z	Relevance x	 y	Plausibility z	 y
3-shot prompting	GPT-3	76	92	97	100	97	74	75
	GPT-4	94	91	99	98	95	90	85
SFT with LLM	LlaMA-7B	89	92	98	98	95	89	80
	FlanT5-XXL	80	93	94	96	93	74	58
	GPT-2-XL	88	92	92	97	94	80	83
SED with LLM	GPT-2-XL	78	87	90	94	94	66	77
EoA with LLM	GPT-2-XL	97	94	94	100	94	88	86

Table 11: Fine-grained human evaluation on **un**-RocStories.

Supervision	Model	Consistency x	 y	 z	Relevance x	 y	Plausibility z	 y
Few-shot prompting	GPT-3	93	92	99	100	94	93	84
	GPT-4	98	92	100	99	97	95	90
SF with LLM	LlaMA-7B	93	94	100	99	97	90	88
	FlanT5-XXL	94	91	97	96	96	88	80
	GPT-2-XL	96	95	97	98	98	91	91
SED with LLM	GPT-2-XL	94	84	97	97	85	92	73
EoA with LLM	GPT-2-XL	91	95	97	97	94	87	88

Table 12: Fine-grained human evaluation on **un**-SocialIQA.

5. Is the explanation not contradictory to the outcome? (denoted as consistency y)

6. Is it possible that explanation might occur (given the context)? (denoted as plausibility z)

7. Is the outcome more likely given the context and the explanation than given the context alone? (plausibility y)

The results are presented in Table 11 for **un**-RocStories, Table 12 for **un**-SocialIQA, Table 13 for αNLG, and Table 14 for Sen-Making.

We also compute BERTScore, ROUGE-L, METEOR, SacreBLEU, and BLEURT for each method and report the results in Table 15 for **un**-RocStories, Table 16 for **un**-SocialIQA, Table 17 for αNLG, and Table 18 for Sen-Making.

D Templates

We include the following prompting templates:

- Figure 5: The prompt to combine a question and its answer into a single sentence on **un**-SocialIQA with five demonstrations.
- Figure 6: The prompt to generate improbable answers on **un**-SocialIQA with six demonstrations.
- Figure 7: The prompt to estimate the outcome likelihood given the context.
- Figure 8: The prompt to generate explanations on **un**-SocialIQA with three demonstrations.
- Figure 9: The prompt to generate explanations on **un**-RocStories with three demonstrations.
- Figure 10: The prompt to improve a crowd-written explanation.

We also include the following MTurk templates:

- Figure 11: The template to collect crowd-written explanations.
- Figure 12: The template to collect human preferences.

E Crowdsourcing Details

Tasks which are allocated to a worker but not completed are later distributed to the entire group of workers. We allow workers at least a week to complete each of their allocated tasks, which allows them sufficient time to complete the task and work at their own pace.

E.1 Qualification.

We use a qualification task to recruit and train workers to produce quality explanations of uncommon

		Consistency			Relevance		Plausibility	
Supervision	**Model**	x	y	z	x	y	z	y
Few-shot prompting	GPT-3	100	97	100	99	96	98	94
	GPT-4	99	98	99	99	98	99	97
SFT with LLM	LlaMA-7B	99	97	99	95	97	98	91
	FlanT5-XXL	95	92	95	93	81	96	85
	GPT-2-XL	96	92	98	93	97	94	85
SED with LLM	GPT-2-XL	97	91	99	95	93	96	84
EoA with LLM	GPT-2-XL	97	95	98	97	97	98	90

Table 13: Fine-grained human evaluation on αNLG.

		Consistency	Relevance	Plausibility
Supervision	**Model**	y	y	y
Few-shot prompting	GPT-3	100	100	93
	GPT-4	100	100	99
SFT with LLM	LlaMA-7B	91	95	86
	FlanT5-XXL	85	98	84
	GPT-2-XL	86	97	85
SED with LLM	GPT-2-XL	92	98	91
EoA with LLM	GPT-2-XL	87	95	83

Table 14: Fine-grained human evaluation on Sen-making.

Combine the following question and answer into a sentence: What will Others want to do next? quit their job and start their own business.
Others will want to quit their job and start their own business.

Combine the following question and answer into a sentence: How would you describe Remy? selfish
Remy is selfish.

Combine the following question and answer into a sentence: What will happen to Quinn? they will spontaneously combust
Quinn will spontaneously combust.

Combine the following question and answer into a sentence: How would you describe Bailey? do not want a healthy pet
Bailey does not want a healthy pet.

Combine the following question and answer into a sentence: How would you describe Carson? like Carson was mean
Carson is mean.

Combine the following question and answer into a sentence: {question} {answer}

Figure 5: Prompting template for combining a question and its answer.

Context: Sydney walked past a homeless woman asking for change but did not have any money they could give to her. Sydney felt bad afterwards.
Question: How would you describe Sydney?
An unlikely answer: ridiculous

Context: Jesse set Robin's suitcase on fire after their fight and messy breakup.
Question: What will Jesse want to do next?
An unlikely answer: decide not to reconcile

Context: Bailey asked Sasha's grandma about church because they wanted to know more about it.
Question: What will happen to Sasha?
An unlikely answer: they get yelled by Sasha's grandma

Context: Bailey told Alex to send the letter overnight since it was important.
Question: How would Alex feel as a result?
An unlikely answer: rushed

Context: Lee made copies so that everyone at the table could follow along.
Question: What will Lee want to do next?
An unlikely answer: ask people stop reading the paper

Context: Taylor gave Kai a lot to think about.
Question: What will happen to Kai?
An unlikely answer: not talk to Taylor

Context: {context}
Question: {question}
An unlikely answer:

Figure 6: Prompting template for generating improbable answers for SocialIQA examples.

Supervision	**Model**	BERTScore	ROUGE	METEOR	SacreBLEU	BLEURT
3-shot prompting	GPT-4	90.79	30.43	29.79	5.16	-21.74
SFT with LLM	GPT2-XL	90.01	26.67	24.17	3.44	-35.22
SED with LLM	GPT2-XL	89.76	26.08	22.64	2.94	-40.04
EaO with LLM	GPT2-XL	89.91	26.79	25.72	3.94	-30.32

Table 15: Automatic evaluation on **un**-RocStories.

Supervision	**Model**	BERTScore	ROUGE	METEOR	SacreBLEU	BLEURT
3-shot prompting	GPT-4	90.79	30.43	29.79	5.16	-21.74
SFT with LLM	GPT2-XL	90.01	26.67	24.17	3.44	-35.22
SED with LLM	GPT2-XL	89.76	26.08	22.64	2.94	-40.04
EaO with LLM	GPT2-XL	89.91	26.79	25.72	3.94	-30.32

Table 16: Automatic evaluation on **un**-SocialIQA.

A: {context}
B: {outcome}
On the scale from 1 to 5, how likely is B given A?

Figure 7: Prompting template for estimating the likelihood of the outcome given the context.

Context: Cameron decided to have a barbecue and gathered her friends together.
Outcome: Others feel bored and uninterested.
Explanation of the outcome: Other than eating the food, there weren't other activities planned.

Context: Jan needed to give out jobs for an upcoming project at work.
Outcome: Others will take a nap instead of working.
Explanation of the outcome: The others don't get paid more for doing the jobs Jan gave out.

Context: Remy was an expert fisherman and was on the water with Kai. Remy baited Kai's hook.
Outcome: Remy will eat a sandwich.
Explanation of the outcome: It's been too long before they feel the weight of a fish, and Remy is hungry.

Context: {context}
Outcome: {outcome}
Explanation of the outcome:

Figure 8: Prompting template for generating explanations for **un**-SocialIQA examples.

Context: My friends all love to go to the club to dance. They think it's a lot of fun and always invite. I finally decided to tag along last Saturday. I danced terribly and broke a friend's toe.
Outcome: My friends decided to keep inviting me out as I am so much fun.
Explanation of the outcome: My friends thought the way I dance is really funny and they couldn't stop laughing.

Context: On the fourth of July, Lilly baked a lemon blueberry cake. She brought it to her boyfriend's house and they had a bbq. After dinner they drove into the city to watch fireworks. When the show was over, they got donuts from a food truck.
Outcome: Lilly had a terrible date.
Explanation of the outcome: Lilly's boyfriend kept complaining that the donuts were way better than the lemon blueberry cake she made, and her boyfriend just threw the cake away.

Context: Jennifer was bored one Saturday. She decided to alleviate her boredom with a hike. She drove to a national park to go hiking. Jennifer hiked for hours.
Outcome: Jennifer thought hiking was stupid.
Explanation of the outcome: She realized the Saturday was a holiday, and the hiking trails in the national park were too crowded that it took her longer than usual to finish.

Context: {context}
Outcome: {outcome}
Explanation of the outcome:

Figure 9: Prompting template for generating explanations for **un**-RocStories examples.

Supervision	**Model**	BERTScore	ROUGE	METEOR	SacreBLEU	BLEURT
3-shot prompting	GPT-4	92.49	34.31	41.58	6.04	-31.02
SFT with LLM	GPT2-XL	92.45	33.52	37.13	6.35	-39.81
SED with LLM	GPT2-XL	92.41	33.39	37.05	6.37	-39.40
EaO with LLM	GPT2-XL	92.17	32.21	37.36	5.81	-38.75

Table 17: Automatic evaluation on αNLG.

Supervision	**Model**	BERTScore	ROUGE	METEOR	SacreBLEU	BLEURT
3-shot prompting	GPT-4	91.40	32.94	47.80	4.99	-16.01
SFT with LLM	GPT2-XL	92.00	36.74	47.28	6.97	-18.00
SED with LLM	GPT2-XL	91.89	36.22	46.58	5.80	-18.57
EaO with LLM	GPT2-XL	91.89	36.07	47.41	5.88	-16.68

Table 18: Automatic evaluation on Sen-Making.

Can you improve this explanation so that it becomes more specific to the context and makes the outcome more likely to happen?

Context: {context}
Outcome: {outcome}
Explanation for the outcome:{explanation}

Figure 10: Prompting template for improving an explanation.

outcomes. In the qualification task, each worker is asked to write an explanation for five pre-chosen contexts paired with uncommon outcomes, including one pair chosen as an attention check. Three paper authors manually grade the explanations to check if they make the outcomes more likely, naturally follow the contexts, and leave little information gaps in-between. We qualify workers who provide at least three high-quality explanations, resulting in qualifying 204 out of 520 workers.

E.2 Quality Control for Crowd-written Explanations.

To ensure the quality of crowd-written explanations, we maintain active communication with workers, and detect and filter low-quality explanations. We engage with workers through an online group chat and periodically provide personalized feedback to individual workers. We detect low-quality explanations through multiple manual and automatic filters, e.g., checking for contradictions between the worker-written explanation and the context and outcome. We dequalify 22 workers who contribute more than two low-quality out of five randomly sampled explanations, and remove all of their explanations from the dataset.

Additionally, we have following automatic ways to verify workers' explanations:

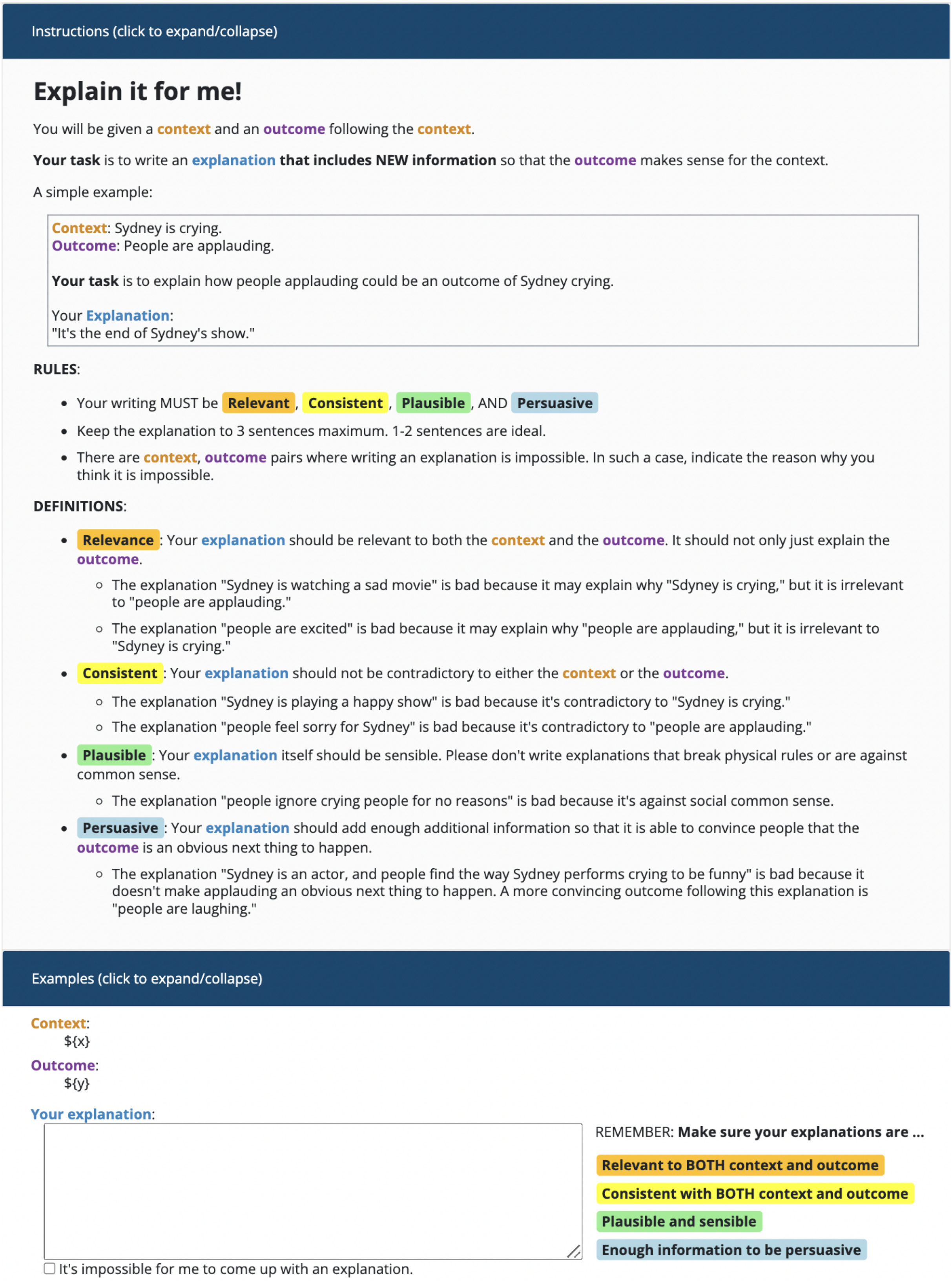
- We use GPT3 to check contradiction between a context and its corresponding explanations.
- We use GPT2 to check relevance to the context via $p(y|x,z) - p(y|z) > \epsilon$.
- In each launch, we sample one explanation from each worker, and we send individual feedback to the workers who violate our rules and filter out the workers who contributed bad explanations to us
- We check how many examples are marked impossible to explain by each worker, and remove workers who use such marks too often.

F Experiment Model Details

We implement both the baseline and the proposed approaches with Hugging Face Transformers (Wolf et al., 2020). We train all models with a learning rate of 0.00001 and a batch size of 8. We perform grid search with $\lambda \in \{1, 0.1, 0.01\}$ and $\beta \in \{0.1, 0.01, 0.001\}$, and we choose the best performing checkpoint on the development set. In DAgger, we set epochs I to be five and block size k to be 2 tokens.

G Static expert demonstrations pseudo-code

The pseduocode for the static expert demonstrations algorithm introduced in §4.2 is given in Algorithm 2.



Instructions (click to expand/collapse)

Explain it for me!

You will be given a **context** and an **outcome** following the **context**.

Your task is to write an **explanation** **that includes NEW information** so that the **outcome** makes sense for the context.

A simple example:

Context: Sydney is crying.
Outcome: People are applauding.

Your task is to explain how people applauding could be an outcome of Sydney crying.

Your **Explanation**:
"It's the end of Sydney's show."

RULES:

- Your writing MUST be **Relevant**, **Consistent**, **Plausible**, AND **Persuasive**
- Keep the explanation to 3 sentences maximum. 1-2 sentences are ideal.
- There are **context**, **outcome** pairs where writing an explanation is impossible. In such a case, indicate the reason why you think it is impossible.

DEFINITIONS:

- **Relevance** : Your **explanation** should be relevant to both the **context** and the **outcome**. It should not only just explain the **outcome**.
  - The explanation "Sydney is watching a sad movie" is bad because it may explain why "Sdyney is crying," but it is irrelevant to "people are applauding."
  - The explanation "people are excited" is bad because it may explain why "people are applauding," but it is irrelevant to "Sdyney is crying."
- **Consistent** : Your **explanation** should not be contradictory to either the **context** or the **outcome**.
  - The explanation "Sydney is playing a happy show" is bad because it's contradictory to "Sydney is crying."
  - The explanation "people feel sorry for Sydney" is bad because it's contradictory to "people are applauding."
- **Plausible** : Your **explanation** itself should be sensible. Please don't write explanations that break physical rules or are against common sense.
  - The explanation "people ignore crying people for no reasons" is bad because it's against social common sense.
- **Persuasive** : Your **explanation** should add enough additional information so that it is able to convince people that the **outcome** is an obvious next thing to happen.
  - The explanation "Sydney is an actor, and people find the way Sydney performs crying to be funny" is bad because it doesn't make applauding an obvious next thing to happen. A more convincing outcome following this explanation is "people are laughing."

Examples (click to expand/collapse)

Context:
${x}

Outcome:
${y}

Your explanation:

REMEMBER: **Make sure your explanations are ...**

Relevant to BOTH context and outcome

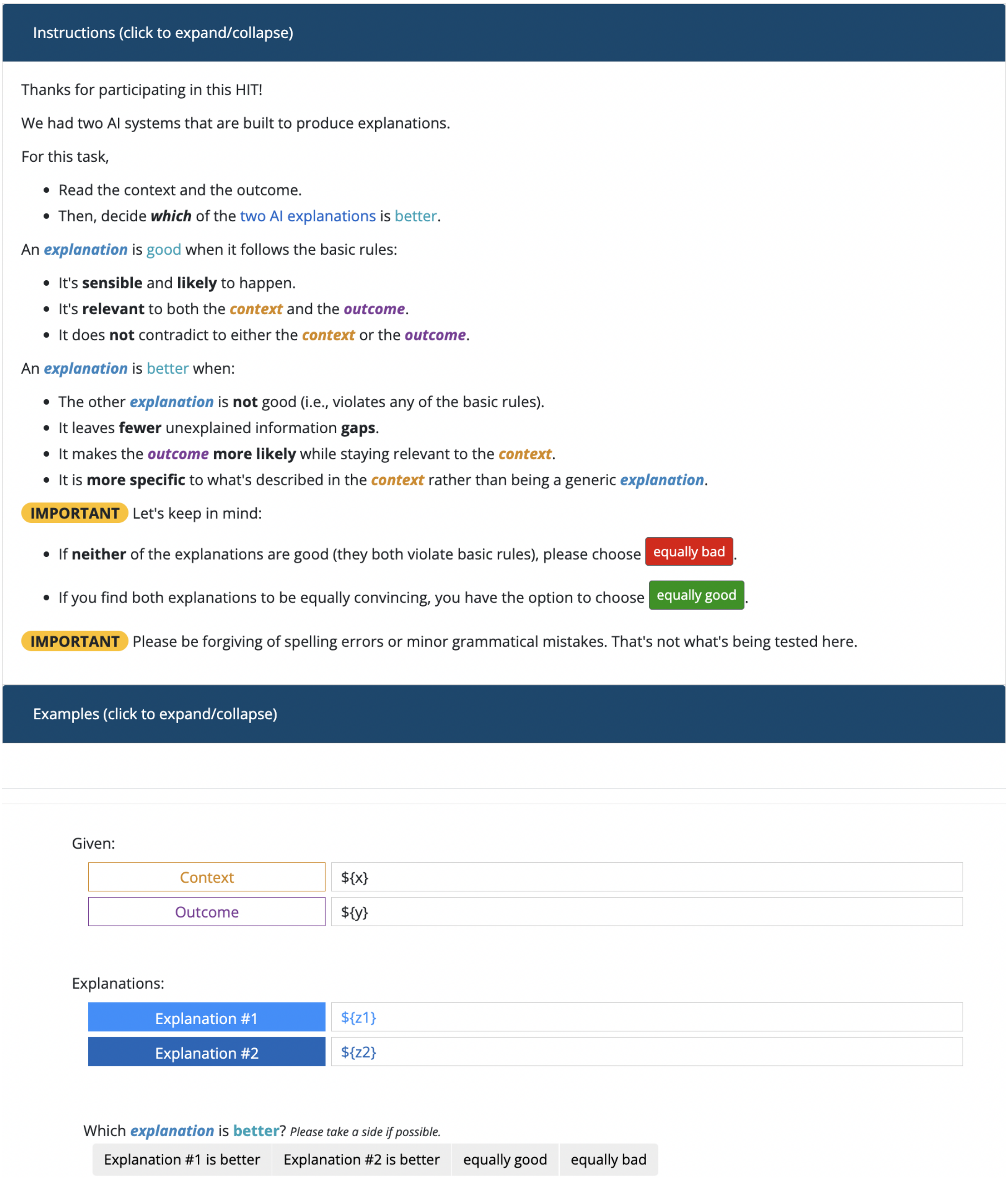
Consistent with BOTH context and outcome

Plausible and sensible

Enough information to be persuasive

□ It's impossible for me to come up with an explanation.

Figure 11: A screenshot of mturk template for collecting explanations.



Instructions (click to expand/collapse)

Thanks for participating in this HIT!

We had two AI systems that are built to produce explanations.

For this task,

- Read the context and the outcome.
- Then, decide ***which*** of the two AI explanations is better.

An ***explanation*** is good when it follows the basic rules:

- It's **sensible** and **likely** to happen.
- It's **relevant** to both the ***context*** and the ***outcome***.
- It does **not** contradict to either the ***context*** or the ***outcome***.

An ***explanation*** is better when:

- The other ***explanation*** is **not** good (i.e., violates any of the basic rules).
- It leaves **fewer** unexplained information **gaps**.
- It makes the ***outcome*** **more likely** while staying relevant to the ***context***.
- It is **more specific** to what's described in the ***context*** rather than being a generic ***explanation***.

IMPORTANT Let's keep in mind:

- If **neither** of the explanations are good (they both violate basic rules), please choose equally bad.
- If you find both explanations to be equally convincing, you have the option to choose equally good.

IMPORTANT Please be forgiving of spelling errors or minor grammatical mistakes. That's not what's being tested here.

Examples (click to expand/collapse)

Given:

Context	${x}
Outcome	${y}

Explanations:

Explanation #1	${z1}
Explanation #2	${z2}

Which ***explanation*** is **better**? *Please take a side if possible.*

Figure 12: A screenshot of mturk template for doing pair-wise preference evaluation.

Algorithm 2 Online learning with static expert demonstrations.

```
Inputs: Initial learner policy parameters $\theta_0$, dataset $\mathcal{D} = \{(x, y, z)\}^N$, number of training epochs $I$.
$\tilde{D} \leftarrow \emptyset$
for $i = 0, \ldots, I - 1$ do
    for $(x, y, z) \in \mathcal{D}$ do
        $\tilde{z} \sim \pi(. \mid x, y)$
        $\tilde{\mathcal{D}} \leftarrow \tilde{\mathcal{D}} \cup \{(x, y, z, \tilde{z})\}$
    end for
    $\theta_{i+1} \leftarrow \theta_i$ further optimized on $\tilde{D}$ using the objective in Equation 1.
end for
Returns: Learned policy parameters $\theta_I$.
```