# ClarifyDelphi: Reinforced Clarification Questions with Defeasibility Rewards for Social and Moral Situations

**Valentina Pyatkin**[♢♠] **Jena D. Hwang**[♠] **Vivek Srikumar**[♠♣] **Ximing Lu**[♡♠]
**Liwei Jiang**[♡♠] **Yejin Choi**[♡♠] **Chandra Bhagavatula**[♠]
[♢]Bar-Ilan University [♣]University of Utah
[♠]Allen Institute for Artificial Intelligence
[♡]Paul G. Allen School of Computer Science & Engineering, University of Washington
pyatkiv@biu.ac.il

## Abstract

*Context is everything*, even in commonsense moral reasoning. Changing contexts can flip the moral judgment of an action; *Lying to a friend* is wrong in general, but may be morally acceptable if it is intended to protect their life.

We present CLARIFYDELPHI, an interactive system that learns to ask clarification questions (e.g., "why did you lie to your friend?") in order to elicit additional salient contexts of a social or moral situation. We posit that questions whose potential answers lead to *diverging* moral judgments are the most informative. Thus, we propose a reinforcement learning framework with a *defeasibility reward* that aims to maximize the divergence between moral judgments of hypothetical answers to a question. Human evaluation demonstrates that our system generates more relevant, informative and defeasible questions compared to competitive baselines. Our work is ultimately inspired by studies in cognitive science that have investigated the flexibility in moral cognition (i.e., the diverse contexts in which moral rules can be bent), and we hope that research in this direction can assist both cognitive and computational investigations of moral judgments.

## 1 Introduction

Commonsense moral reasoning of social situations and actions depends squarely on their context. *Offering someone a cup of coffee* is generally considered appropriate. If offered to a work colleague, it may even be viewed as a courteous gesture. However, offering coffee to a toddler would be deemed morally irresponsible.

Delphi (Jiang et al., 2022), a recently proposed commonsense moral reasoning model, generates moral judgments for simple actions described in text. However, Delphi's judgments are made in isolation, without any knowledge of surrounding context. Grounding moral reasoning in context is crucial (Talat et al., 2022). How can moral reasoners elicit missing salient context? A natural way to do so is by asking clarification questions.

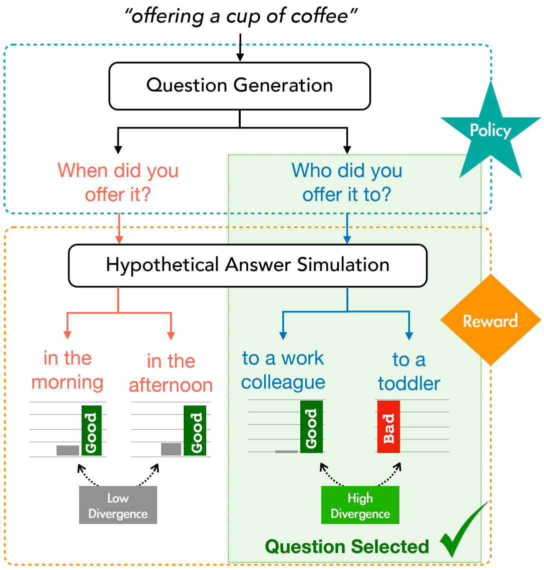


Figure 1: The CLARIFYDELPHI question generation approach is trained via reinforcement learning. The reward simulates a set of possible (defeasible) answers to the questions and, using Delphi for feedback, optimizes for questions leading to maximally diverging answers.

We present CLARIFYDELPHI, an interactive system that learns to ask questions to elicit salient context. Prior research in cognitive science shows that human reasoning exhibits the flexibility not only to articulate where a certain moral rule should hold, but also to imagine valid exceptions where the rule can be bent or *defeated* based on the demands of the context (Kwon et al., 2022; Levine et al., 2020; Awad et al., 2022).

We present a first step toward computationally exploring and discovering these *defeasible* contexts which can potentially flip the moral judgement of a situation. Given a situation and its default judgment (e.g., it is *nice* to offer a cup of coffee to someone), defeasible contexts can strengthen (e.g., *offering it to a colleague*) or weaken (e.g., *giving it*

*Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*
*Volume 1: Long Papers*, pages 11253–11271
July 9-14, 2023 ©2023 Association for Computational Linguistics

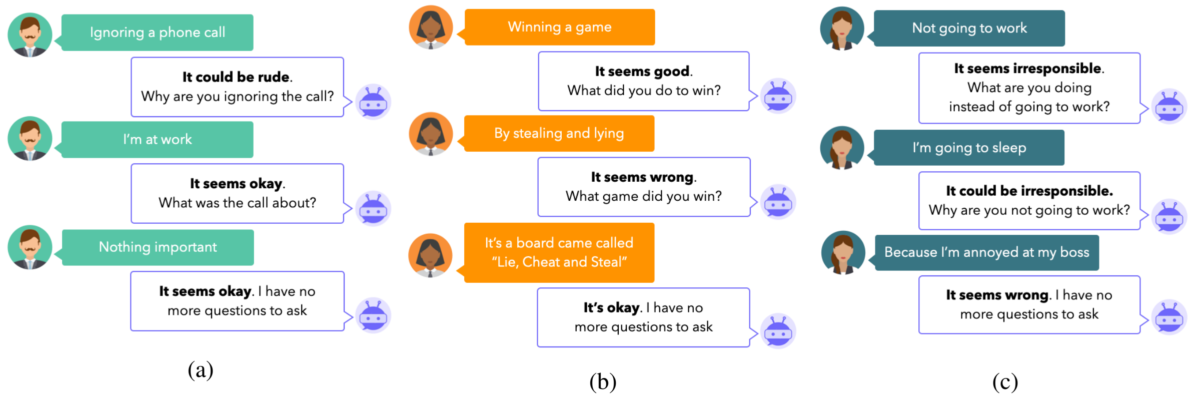

Ignoring a phone call
It could be rude. Why are you ignoring the call?
I'm at work
It seems okay. What was the call about?
Nothing important
It seems okay. I have no more questions to ask
(a)
Winning a game
It seems good. What did you do to win?
By stealing and lying
It seems wrong. What game did you win?
It's a board came called "Lie, Cheat and Steal"
It's okay. I have no more questions to ask
(b)
Not going to work
It seems irresponsible. What are you doing instead of going to work?
I'm going to sleep
It could be irresponsible. Why are you not going to work?
Because I'm annoyed at my boss
It seems wrong. I have no more questions to ask
(c)

Figure 2: Interaction between a user and CLARIFYDELPHI. The user inputs a situation and CLARIFYDELPHI answers with an initial judgement (obtained from DELPHI) and a clarification question, which the user then answers.

*to a toddler*) the judgment (Rudinger et al., 2020; Madaan et al., 2021; Allaway et al., 2022). We aim to generate questions whose answers might uncover missing context for making better-informed moral judgments, and we propose to do so in a conversational setting between a user and CLARIFYDELPHI.

Our method for clarification question generation is based on reinforcement learning. Using Proximal Policy Optimization (PPO; Schulman et al. 2017; Ouyang et al. 2022) we optimize for generating questions that invoke responses that provide morally salient contexts. CLARIFYDELPHI "imagines" answers to a generated question, using a trained answer generation model. A reward is calculated by comparing the probability distributions Delphi assigns to the imagined answers. Fig. 1 provides an overview of CLARIFYDELPHI.

The intuition behind our approach is that questions that lead to maximally divergent answers (e.g., "Who did you offer it to?") are also those that elicit most morally salient contexts and therefore are more *consequential* to the situation. These morally consequential questions surface latent ambiguities that may directly affect the moral decision process. Questions with little divergence in its imagined answers (e.g., "When did you offer it?") have little to offer in terms of resolving contextual moral ambiguities.

Our results show that our approach outperforms other strong clarification question generation baselines; its generated questions lead to consequential answers. We additionally quantify how much supervised clarification question training data is needed for a good initial policy. Lastly we show that questions help with generating defeasible updates.

Our contributions are as follows. We introduce the task of clarification question generation for social and moral situations. For this task we propose an RL based approach, defining defeasibility as a new type of relevance for clarification questions. We publicly release $\delta$-CLARIFY, a dataset of 33k crowdsourced clarification questions, and $\delta$-CLARIFY $_{silver}$ containing generated questions conditioned on a defeasible inference dataset. We also release trained models with their code.[1]

## 2 Problem Setup

Given a situation, such as *lie to my friend*, we aim to generate question(s) that are the most relevant for uncovering the most consequential context with respect to making a social or moral judgement. While situations could evoke a multitude of potential questions, the following work is concerned with predicting questions whose answers are likely to be *consequential*, i.e. answers that could function as either weakeners or strengtheners of the default judgement. The terms *weakener* and *strengthener* come from the concept of defeasible inference (Rudinger et al., 2020), which defines a way of reasoning that takes into consideration (new) evidence which could either support (e.g. *strengthen*) or cancel/*weaken* an initial inference.

Formally, the task is to predict a question $q$ given a base situation $s$. The base situation has a default moral judgement $j \in \{bad, ok, good\}$. For every input tuple of $(s_i, q_i, j_i)$ there is a hypothetical set of strengthening answers $A_S$ and weakening answers $A_W$. Adding the additional information obtained from any $q_i$ and corresponding answer $a_i$ to

[1] Data and code are available at: https://github.com/allenai/clarifydelphi.

**Algorithm 1** Training ClarifyDelphi

```
Input initial policy model $\theta_0$, initial value model $\phi_0$, Delphi $\psi_{\text{Delphi}}$
  $D_{\delta\text{-CLARIFY}} \leftarrow$ Get dataset of clarification questions.
  $\theta_{\text{Q}} \leftarrow$ Fine-tune $\theta_0$ with Eqn 1 from $D_{\delta\text{-CLARIFY}}$.   ▷ Sec. 3.1
  $D_{\delta\text{-CLARIFY}_{silver}} \leftarrow$ Get silver dataset of defeasible answers to questions.
  $\theta_{\text{A}} \leftarrow$ Fine-tune lm with Eqn 2 from $D_{\delta\text{-CLARIFY}}$.   ▷ Sec. 3.2
  $\theta_{\text{CLARIFYDELPHI}} \leftarrow$ REINFORCEDLEARNING($S_{\text{SocialChem}}$, $\theta_{\text{Q}}$, $\theta_{\text{A}}$, $\phi_0$, $\psi_{\text{Delphi}}$)
                                                                  ▷ Sec. 3.4
  procedure REINFORCEDLEARNING($S_{\text{SocialChem}}$, $\theta_{\text{Q}}$, $\theta_{\text{A}}$, $\phi$, $\psi_{\text{Delphi}}$)
      $\theta_{Q_{\text{old}}} \leftarrow \theta_Q$, $\phi_{\text{old}} \leftarrow \phi$
      for iterations = 1, 2, ... do
          Sample a minibatch of situations $s$ from $S_{SocialChem}$.
          for step = 1, 2, ..., $s$ do
              Calculate $r$ using $\theta_A$ and $\psi_{\text{Delphi}}$ with Eqn 3.
              Compute $loss_{\text{PPO}}$ on the minibatch with Eqn 6.
              Optimize $\theta$ and $\phi$ with $\mathcal{L}_{\text{PPO}}$ for one step.
          $\theta_{Q_{\text{old}}} \leftarrow \theta_Q$, $\phi_{\text{old}} \leftarrow \phi$
      return $\theta_Q$
Output $\theta_{\text{CLARIFYDELPHI}}$
```

the base situation $s_i$ results in an updated situation $s_{ui}$, with an updated judgement $j_{ui}$.

## 3 ClarifyDelphi: A Reinforced Clarification Question Generator

The ClarifyDelphi approach is based on reinforcement learning. Algorithm 1 gives an overview of the training process. As a first step, before performing reinforcement learning, we obtain a question generation model $\theta_Q$ and an answer generation model $\theta_A$, which we both train on data that we curated, described in the later Sec. 4. The question generation model predicts the clarification questions and the answer generation model provides (defeasible) answers to the generated questions. By using these two models in addition to Delphi ($\psi_{Delphi}$) for calculating the rewards, we do not require any supervised data during RL training.

We consider question generation conditioned on a given situation a sequential decision making process over the natural language vocabulary space, where the generated question $q$ with $T$ tokens has an episode of length $T$. At step $t \in [1, T]$, the state $s_t = (s, q_{<t})$ is the combination of the given situation and the question decoded up to the $(t-1)$-th token; the action $c_t = q_t$ would be the $t$-th token to decode. The question generation model, $\theta_Q(q_t|q, q_{<t}; \theta)$, is the *policy model* that we optimize. We define a reward function $r(s, q, a_w, a_s)$ that characterizes the divergence of answers from $\theta_A$ conditioned on generated question $q$ and discuss the definition of this reward function in §3.3.

### 3.1 Supervised Question Generation

The first subcomponent is a basic question generation system $\theta_Q$ that outputs a question $q$ conditioned on a situation $s$. It is used as the initial policy model during RL training.

$$\hat{q} = \underset{q}{arg\,max} P(q|s) \tag{1}$$

### 3.2 Defeasible Answer Simulation

For each generated question $q$, we need to generate a weakening answer $a_w$ and a strengthening answer $a_s$ in order to calculate the reward $r$ (Formula 3). For the defeasible answer generation system $\theta_A$, we take as input a situation $s_i$, the generated question $q_i$ (§3.1), and an update type $u \in \{weakener, strengthener\}$ to predict a weakener-strengthener answer pair $a_w$ and $a_s$:

$$a = \underset{a}{arg\,max} P(a|s, q, u) \tag{2}$$

An example of an instantiated input/output:

**Input** It's bad to be a snitch, TYPE: Weakener, Q.: Why would being a snitch be beneficial?
**Output** doing so would save someones life.

The crucial element in the input is the update type, as it allows to generate two types of answers for the same $s$ and $q$. When computing the reward during training, for each question, we filter out all its generated answers which either contradict or are entailed (i.e. no new information) by the given situation, using an off-the-shelf NLI model.

### 3.3 Reward

As a reward for generating a question, we aim to quantify how well the generated questions are able to elicit consequential answers. For this purpose we query Delphi (Jiang et al., 2022) for feedback, using situations updated with answers.

We optimize for questions that lead to maximally divergent answers by defining a reward function which uses the JS-Divergence, between the Delphi probability distribution of the weakener updated situation and the strengthener updated situation:

$$r(s, q, a_w, a_s) = JSD(P_{jw}||P_{js}) \tag{3}$$

**Sentence Fusion** To create an *updated situation* that sounds natural and can be used to query Delphi, the situation $s$, question $q_i$ and answer (both $a_w$ and $a_s$ separately) have to be fused together into $s_{ui}$. For example:

**Situation** refraining from doing something bad
**Question** When do you do something bad?
**Answer** when I'm angry
**Fusion:** *refraining from doing something bad when you're angry.*

We train a model to distill fusion in-context examples obtained from GPT-3 (text-curie-001).

**Delphi for Feedback** Delphi is then queried with the updated situation $s_{ui}$ for a judgement, leveraging the probability distribution that Delphi provides over three classes: $j \in \{\text{bad}, \text{ok}, \text{good}\}$. The probability scores are the probabilities of the special T5 tokens representing each of the three classes, normalized by their sum.

$$j = \underset{q}{arg\,max_j} P(j|s) \tag{4}$$

**JS-Divergence** We calculate the Jensen-Shannon divergence between the Delphi probability distributions $j_w$ and $j_s$ obtained from two updated situations originating from defeasible answers to $q$.

**Reward normalization** We normalize the reward during training as follows:

$$r(x, k) \leftarrow \frac{r(x, k) - \mu_0}{\sigma_0}. \tag{5}$$

The $\mu_0$ and $\sigma_0$ of rewards are obtained before training begins, by generating a question and calculating its rewards for all $s$ in the training data.

### 3.4 Proximal Policy Optimization (PPO)

We maximize the reward using Proximal Policy Optimization (PPO) (Schulman et al., 2017) as our RL algorithm, which previous works have shown to be suitable for NLG tasks (Liu et al., 2022b; Ramamurthy et al., 2022). Our implementation of PPO is an adaptions of Ouyang et al. (2022)'s, which includes a KL penalty between the initial policy model $\theta_{Q_{old}}$ and the updated policy $\theta_Q$. In addition to the policy model, PPO employs a *value model* (parameterized by $\phi$) to estimate the value function for states with incomplete decoded text, i.e. $V(s_t; \phi)$ for any $t$. PPO's loss consists of a value model ($\phi$) loss and the policy loss, which is jointly minimized:

$$loss_{\text{PPO}}(\theta, \phi) = \alpha \cdot loss_{\text{Value}}(\phi) + loss_{\text{Policy}}(\theta) \tag{6}$$

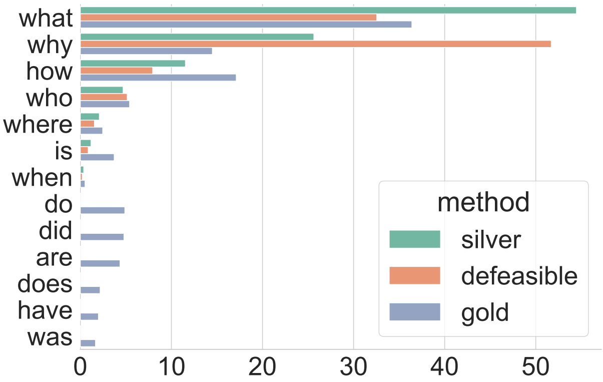


Figure 3: Proportional distribution (%) of the most frequent question starts in $\delta$-CLARIFY$_{gold}$, $\delta$-CLARIFY$_{silver}$ and the subset of *defeasible* questions of $\delta$-CLARIFY$_{silver}$.

## 4 $\delta$-CLARIFY: a Dataset of Clarification Question

We require data for various components of our CLARIFYDELPHI model: The policy needs bootstrapping from a clarification question dataset and the answer generation model needs data to learn to generate defeasible answers to questions. To the best of our knowledge no such datasets exist. We therefore collect a crowdsourced dataset of clarification question for social and moral situation and a silver dataset of defeasible QAs to train $\theta_Q$ .

The situations are sampled from SOCIAL-CHEM-101 (Forbes et al., 2020) and the COMMONSENSE NORM BANK (Jiang et al., 2022). We call our dataset $\delta$-CLARIFY and it consists of crowdsourced questions, enriched with questions generated by GPT-3 (Brown et al., 2020).

Next we describe how we create the dataset.

| *Situation:* | **Tipping people decently** |
|---|---|
| Q1 | What did they do for you? |
| Q2 | Can you afford to tip? |
| Q3 | Was the service good? |
| Q4 | Did the people perform the service adequately? |
| Q5 | Do you always tip people well regardless of the service quality? |
| *Situation:* | **Jeff ignores the comment and laughs about it with his boss** |
| Q1-4 | What was the comment? |
| Q5 | Who made the comment they were laughing at? |

Table 1: Two examples of situations and their clarification questions, written by five different Turkers.

**$\delta$-CLARIFY$_{gold}$:** We crowdsource clarification questions by showing annotators a situation and asking them to write a clarification question they would ask an imagined colleague requesting advice on the situation. Each of the 6425 situations is presented to 5 annotators, resulting in 5 questions per situation (500 situations are used for dev and test respectively). Details of the annotation are found in Appendix A.1.

**$\delta$-CLARIFY$_{silver}$:** The $\delta$-SOCIAL part of the defeasible inference dataset (Rudinger et al., 2020) consists of *statements* that express default judgments over situations (*It is good to protect your kids*) and *updates* that weaken (*Your protection is overbearing*) or strengthen (*Your child is in danger*) the default. These updates could be viewed as potential answers to an implicit question about a base situation: "What are you protecting your child from?" for *Your child is in danger.* We 5-shot prompt GPT-3 to generate questions, conditioned on situation and answer, resulting in $\approx\ 80K$ $(situation, update\,type, question, answer)$ tuples.

**Dataset Analysis** Fig. 3 shows that the crowdsourced $\delta$-CLARIFY$_{gold}$ has more variety in its most common question starts, which reflects the general diversity of the dataset: For only 10% of the situations, more than 1 Turker asked exactly the same question, and for only 8% of the situations all 5 Turkers used the same Wh-word to start the question. This highlights that there is more than one possible salient clarification question to be asked for any given situation. For the situation *tipping people decently* in Tab. 1, all 5 Turkers chose to start their questions differently, even though three out of these five questions ask in one way or the other about the service quality. For the other situation 4/5 Turkers asked for a specification "What was the comment?" and 1 Turker asked about the missing agent role. We also see that polar (yes/no) questions appear less frequently, as Turkers were explicitly asked to avoid them unless no other suitable question comes to mind.[2]

The $\delta$-CLARIFY$_{silver}$ questions are generated by conditioning on weakener or strengthener updates. Since we aim to predict *defeasible* questions, the most desirable questions are those whose answers can be both weakeners and strengtheners. In the silver data, 53% of situations have at least one question that has been generated by GPT-3 for both update types. The situation *Your kids should be your number one priority*, for example, has the same question "What are your kids' ages?" for the weakener update *They are adult children.* and the strengthener update *Your children are toddlers.* Interestingly, among the subset of *defeasible* questions in $\delta$-CLARIFY$_{silver}$, we find that the most frequent question start is 'why'. This suggests that it is easiest to come up with both weakener and strengthener answers to why-questions.

## 5 Baselines

We consider four baselines in our experiments.

**Question Generation Without RL** To assess what additional improvements training an RL model with a defeasibility rewards provides, we report performance of the supervised question generation model $\theta_Q$ on its own (§3.1). We refer to this baseline as *t5 fine-tuned*. We decode using nucleus sampling with top-$p = 0.6$.

**Pipelines with Question Selection** Next, we implement two pipelines where, as the first step, a diverse set of questions is generated for a given situation and then, as the second step, the best question is selected according to a score.

In order to generate a diverse set of questions we fine-tune T5 on $\delta$-CLARIFY, conditioned on a modified input compared to the model from §3.1: **Input** <Situation>. Q.: <wh-word> - **Output** <Question>

By also conditioning on the first wh-word of the question it is possible to generate different questions. During inference we generate questions for 14 different question starts.[3] We propose two approaches to scoring these questions: using a discriminator model and using divergence ranking, which we describe as follows.

**Discriminator** We train a discriminator classifier which labels these questions as either *relevant* or *irrelevant* to a given situation. We then choose the question that has been assigned the *relevant* label with the highest probability.

The discriminator is a binary classifier based on DeBERTa (He et al., 2020). The positive examples are situations and their respective 5 questions written by annotators. The negative question examples

[2] This is to prevent leading questions such as "Do you intend to give it to a kid?" for "offering a cup of coffee".

[3] *what, how who, do, are, did, is where, have, was when, would*

are sampled from other situations, in a way that ensures that the *relevant* and *irrelevant* questions are similar enough to make training more challenging.

**Divergence Ranking** We run the defeasible answer simulation with feedback from Delphi for each question in the set. This process is the same as the reward function of the RL approach, except that the JS-divergence score is used to rank the questions instead of being used as a reward for question generation. We compare two variations of this baseline: one with answer filtering using an NLI model as described in Sec. 3.2 (*pipeline-nli*) and one without filtering (*pipeline*).

**Why-Baseline** We saw in §4 that questions conditioned on weakener/strengthener updates are usually causal questions. Using the same input/output configuration as in the pipeline baseline, we generate a why-question for each situation (called *why*).

# 6 Evaluation and Analysis

## 6.1 Human Evaluation

Automatic evaluation of questions and their usefulness for clarifying moral situations is tricky. While we do have gold reference questions, we have shown that humans will produce diverse questions for the same situation (§4) and just because a question does not appear in the reference set does not necessarily indicate that it is not a consequential question. We therefore perform human evaluation of the models' outputs on Amazon Mechanical Turk on the 500 test set instances from $\delta$-CLARIFY. Given a situation and a question, Turkers are asked to rate the question along three different attributes: **Grammaticality** (Is the question grammatical?), **Relevance** (Is the question relevant and plausible to the situation?), and **Informativeness** (Does the question access new information or regurgitate the situation?). The attributes are further detailed in Appendix A.1.

Additionally, and most importantly, we aim to evaluate the **defeasibility** of the questions, e.g. how well the generated questions can elicit *weakener* or *strengthener* answers. For this purpose, Turkers are given a situation with a question and are first asked to judge this situation (*generally ok* vs. *generally not ok*). They then need to say whether and specify if they can think of an answer to the question which might *support* their judgement and also of an answer which would *flip* their judgement.

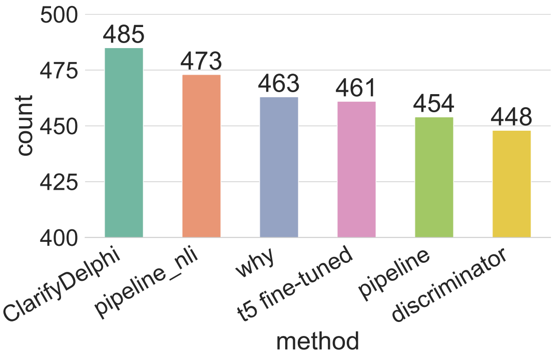


Figure 4: Number of questions (out of 500) in test set that received an *informativeness* and *relevance* rating of $> 0$.

## 6.2 Results of Human Evaluation

We first run the *grammaticality*, *relevance* and *informativeness* evaluation. All questions which are given the lowest rating (e.g. *irrelevant* and/or *uninformative*) by at least two annotators are excluded from the second evaluation. It does not make sense to ask about defeasibility for questions which already are *irrelevant*, and additional weakening or strengthening context is not feasible for *uninformative* questions.

We find, as displayed in Fig. 4, that CLARIFYDELPHI has the biggest percentage of *relevant* and *informative* questions in the test set, compared to the baselines. We also see that a big majority of the generated questions, from all models, are *relevant* and *informative*, with the lowest performing model (*discriminator*) still producing 448/500 questions that are passed on to the next evaluation round.

We also find that *grammaticality* across all systems is high with the lowest average score being 0.98 and the highest 0.99 (on a scale from 0 to 1, with 1 being grammatical). The minimal variation in grammaticality score is expected since all models are based upon the same transformer model.

The CLARIFYDELPHI questions also outperform the baselines in terms of defeasibility, as seen in Table 2: annotators can more often think of a *strengthener* answer and/or a *weakener* answer to our questions. The evaluation also shows that adding the answer-filtering with NLI step to the pipeline improves the question selection on all 4 evaluation criteria. The why-baseline is shown to be a strong baseline, indicating that motives and reasons are important for moral reasoning.

| | defeasibility | weakener | strength. |
|---|---|---|---|
| ClarifyDelphi | **0.44** | **0.47** | **0.73** |
| why | 0.37 | 0.41 | 0.60 |
| pipeline_nli | 0.35 | 0.37 | 0.64 |
| t5 fine-tuned | 0.34 | 0.37 | 0.54 |
| discriminator | 0.33 | 0.36 | 0.55 |
| pipeline | 0.30 | 0.34 | 0.53 |

Table 2: Defeasibility scores obtained through human evaluation. *weakener*: Can you think of an answer to the question that weakens your initial judgement? *strengthener*: Can you think of an answer ot the question that strengthens your intial judgement? *defeasibility*: Can you think of both?.

### 6.3 How much supervision does the policy require?

Our approach uses RL in conjunction with a supervised policy that has been fine-tuned on question generation. This has been shown to outperform approaches which use RL on top of a “vanilla" lm-policy (Ramamurthy et al., 2022). To assess the effect of supervision on question generation performance, we trained multiple initial policies on varying percentages of $\delta$-Clarify training data: 25%, 50%, 75% and 100%. To compare against more traditional supervised question generation approaches we additionally trained a policy on SQuAD v1.1 data (Rajpurkar et al., 2016).

We report two automatic evaluation metrics. To measure *informativeness* we use an off-the-shelf QA model trained on SQuAD 2.0 from AllenNLP (Gardner et al., 2018). This model either answers a question by pointing to a span in the input or outputs that the question in unanswerable with respect to a given context. For a clarification question to be informative it would not ask about anything already mentioned in the situation. For the *QA*-metric we thus report the percentage of non-answerable questions.[4] We also report the average maximum BERTScore (Zhang et al., 2019) between a generated question and one of the 5 gold questions in $\delta$-Clarify.

Fig. 5 shows the following trends with regards to training a supervised policy. More training data leads to more informative questions. The policy trained on SQuAD produces the most uninformative questions which can be explained by the fact that SQuAD questions are conditioned on existing answers in a text. While performance consistently increases from 25% to 75% of the training data, improvements after 75% are minimal. We conclude that for our use case training on about 5000 (75%) situations with 5 questions each leads to a sufficiently good policy. These results are also supported by the BERTScore.

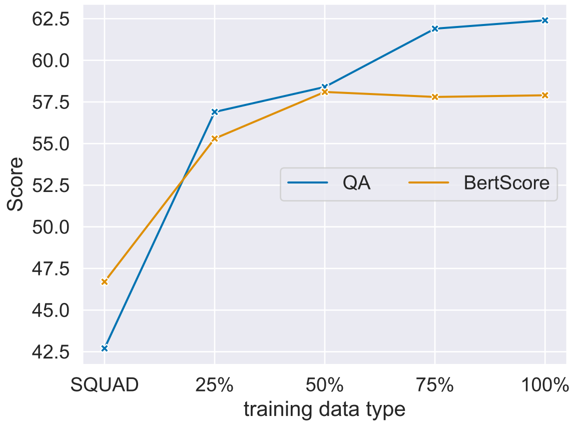


Figure 5: Performance of ppo algorithm with different policies: a policy pre-trained on SQUAD and policies pre-trained on different subsets of the $\delta$-Clarify dataset. The scores (higher is better) are averaged every 1000 steps, between 1000 and 6000.

### 6.4 Analysis

**Answer Simulation** The answer generation model generally succeeds at generating diverse *weakener* and *strengthener* answers to the same question: for only about 0.05% of questions per 1000 PPO epochs the model generates the same answer for both weakener and strengthener.

Our answer generation could be looked at as question-guided defeasible update generation. Rudinger et al. (2020)’s task of *Generative Defeasible Inference* generates an update given a situation, a moral judgement and the update type (e.g. weakener/strengthener). In our answer generation approach we condition on the same input together with a generated question. This intermediate question generation step functions as a type of macro planning which has been shown to be effective for NLG (Puduppully et al., 2022; Narayan et al., 2022). We evaluate our approach on the same test set using the same evaluation metrics as Rudinger et al. (2020). Table 3 shows that by first predicting the question and then the updates, we improve upon generating defeasible updates for $\delta$-social.

**Questions** We qualitatively inspect the types of generated questions: There are many specification questions asking about a hyponym of an argument

[4] The Pearson correlation coefficient shows that this metric (moderately) correlates with the human informativeness evaluation ($r = 0.31$).

| | BLEU | ROUGE |
|---|---|---|
| $\delta$-social (T5-large) | 4.22 | 14.94 |
| $\delta$-social (GPT2-XL) | 12.16 | 18.77 |
| ClarifyD (T5-large) | **14.18** | **34.65** |

Table 3: Automatic results for strengthener/weakener update generation on the $\delta$-social test set. Following Rudinger et al. (2020) we report BLEU-4 (Papineni et al., 2002) and ROUGE-L (Lin and Hovy, 2002). The first two results are from Rudinger et al. (2020).

in the base situation, for example, *exterminating pests on your property* - "What kind of pests?". The situations extracted from SocialChem often include underspecified pronouns, such as 'something' or 'somewhere'. 60% of the situations containing 'something', for example, elicit *what*-questions from our model. Note that while such specification questions are valid clarification questions, the SQUAD 2.0 QA model would mark them as answerable given the situation. It is also interesting to see that often when a situation has a missing or implicit semantic argument, such as *being anxious sometimes*, ClarifyDelphi inquires about it: "What are you anxious about?" The generated *why*-questions most often ask about the motives of the agent in the situation, such as *Ben tells Maggie that he's traveling alone* - "Why is Ben traveling alone?". More rarely the model generates questions asking about the viewpoint of the patient: *asking a friend [...] whether your attire is appropriate for an event* - "What is the friend's opinion of your attire?"

**Analysis of Delphi's Probabilities** In Tab. 4 we quantify the JSD of Delphi's judgments. Even though the human evaluation showed that ClarifyDelphi produced the most questions leading to defeasible answers, the JSD and the precentage of judgment flips is higher for the pipeline_nli approach, where we explicitly filter questions to maximize the JSD. Nevertheless, ClarifyDelphi leads to more Delphi judgment flips and higher JSD between answers than the fine-tuned t5 model without RL (and also all other baselines besides the pipeline). This automatic evaluation and the disagreement with the human annotators also reveals that Delphi's probabilities are not always perfectly calibrated and relying too much on a model's output might potentially lead to some error propagation.

| | avg. JSD | Judgment Flips |
|---|---|---|
| ClarifyDelphi | 0.191 | 25% |
| why | 0.159 | 22% |
| pipeline_nli | 0.259 | 33% |
| t5 fine-tuned | 0.144 | 21% |
| discriminator | 0.138 | 21% |

Table 4: Average JSD between $P_{jw}$ and $P_{js}$ of a situation. Judgment Flips: % of answers which led to a flip in Delphi's judgment.

## 7 Interactive Judgements

While we use answer simulation during PPO training, inference only requires a situation as input. The clarification questions can then be used to elicit additional context, in the form of answers, through interaction. Fig. 2 illustrates examples of such an interaction between a user, Delphi as the moral reasoning system and ClarifyDelphi. After each turn, the situation is updated with the user provided context, for which Delphi produces a new decision. We limit the interaction to three turns. This is based on the observation that after the third turn the sentence fusion starts to deteriorate, resulting in less relevant and more repetitive questions. Additionally, we find that the first two questions generally can capture missing contexts that are most central to making moral decisions. We provide more examples of generated questions in the Appendix.

## 8 Related Work

**Question Generation** Clarification question generation has been studied for various domains from image recognition questions to product description questions (Rao and Daumé III, 2018; Majumder et al., 2021; White et al., 2021), defining the goodness of clarification questions along the lines of information theoretic measures such as relevance, informativeness or utility (Rao and Daumé III, 2018; White et al., 2021; Warstadt and Agha, to appear; Rao and Daumé III, 2018, 2019). Most of existing works focus on questions that lead to single true answer, whereas we focus on generating clarification questions based on social situations, defining the relevance and utility of a question in terms of defeasibility. Additionally, we offer a high-quality clarification question dataset for social and moral situation—comprising of more than 30K questions—that breaks the mold from the domain-specificity of previous clarification datasets (Kumar and Black, 2020; Aliannejadi et al., 2019).

Some general question generation approaches have incorporated an RL-based approach. Buck et al. (2018) learn to paraphrase questions with a reward that maximizes the QA answer F1 score. And Rao and Daumé III (2019) optimize a binary utility reward, using *Reinforce* in an adversarial setup for generating clarification questions. In our setup, we use Proximal Policy Optimization (Schulman et al., 2017; Ouyang et al., 2022) with a trained model for feedback as part of the reward.

**Commonsense Moral Reasoning** Delphi (Jiang et al., 2022) is a commonsense moral reasoning model trained on COMMONSENSE NORM BANK, a dataset with 1.7M instances of descriptive knowledge of people's general sense of what's ethically acceptable or not in everyday situations. COMMONSENSE NORM BANK is compiled from five existing large-scale datasets, including SOCIAL CHEMISTRY (Forbes et al., 2020), ETHICS Commonsense Morality (Hendrycks et al., 2021), MORAL STORIES (Emelin et al., 2021), SOCIAL BIAS INFERENCE CORPUS (Sap et al., 2020), and SCRUPLES (Lourie et al., 2021).

Delphi is based on pre-trained UNICORN, a universal commonsense reasoning model, trained on a number of commonsense reasoning tasks. Delphi can predict the ethical judgment given a description of a situation.

## 9 Conclusion

In this work we introduce CLARIFYDELPHI, which generates clarification questions for social and moral situations. We show how a RL approach that optimizes for maximally divergent answers in terms of defeasibility outperforms other clarification question baselines. While we start with a supervised policy, the reward function makes use of already trained models and does not rely on any additional training data. We believe that our questions can be useful for providing more disambiguating context through interaction.

## Limitations

**On Western-centricity** The majority of the crowdworkers producing the source data ($\delta$-Social and Delphi) and $\delta$-CLARIFY were located in the United States. Due to this, the predictions generated by CLARIFYDELPHI are currently limited to representing only the perspectives of western culture (particularly the United States). Overcoming the western-centric bias is a compelling direction for future research.

**On Defeasibility** We rely upon Delphi to produce acceptable judgments given a situation and the modifying context as a measure of defeasibility. We recognize that, however, Delphi is not perfect and is characterized by a variety of limitations such as limited cultural awareness and inconsistent predictions (Jiang et al., 2022). Investigating improved methods for identifying answer divergences that will better capture defeasibility is a topic for future investigation.

## Ethics Statement

Annotations are conducted on Amazon Mechanical Turk (MTurk). We maintain an average pay of $15 per hour for all our crowdsourcing data collection and evaluation tasks. Our crowdsourcing tasks do not collect personal information and are strictly limited to gathering workers' general knowledge. We do not keep any deanonymizing information such as MTurk IDs so that the identity of the workers cannot be directly or indirectly ascertained. Finally, our crowdsourcing task meets the standards for exemptions as human research and has obtained the necessary documentation that deems it exempt through our internal IRB procedures.

Our model is intended to be used for research purposes only and it is not supposed to provide any sort of advice applicable outside of that domain.

## Acknowledgements

We thank our colleagues on the Beaker Team at the Allen Institute for AI for helping with the compute infrastructure. This work was supported in-part by DARPA MCS program through NIWC Pacific (N66001-19-2-4031).

## References

Mohammad Aliannejadi, Hamed Zamani, Fabio Crestani, and W Bruce Croft. 2019. Asking clarifying questions in open-domain information-seeking conversations. In *Proceedings of the 42nd international acm sigir conference on research and development in information retrieval*, pages 475–484.

Emily Allaway, Jena D Hwang, Chandra Bhagavatula, Kathleen McKeown, Doug Downey, and Yejin Choi. 2022. Penguins don't fly: Reasoning about generics through instantiations and exceptions. *arXiv preprint arXiv:2205.11658*.

Edmond Awad, Sydney Levine, Andrea Loreggia, Nicholas Mattei, Iyad Rahwan, Francesca Rossi, Kartik Talamadupula, Joshua Tenenbaum, and Max

Kleiman-Weiner. 2022. When is it acceptable to break the rules? knowledge representation of moral judgement based on empirical data. *arXiv preprint arXiv:2201.07763*.

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.

Christian Buck, Jannis Bulian, Massimiliano Ciaramita, Wojciech Gajewski, Andrea Gesmundo, Neil Houlsby, and Wei Wang. 2018. Ask the right questions: Active question reformulation with reinforcement learning. In *International Conference on Learning Representations*.

Denis Emelin, Ronan Le Bras, Jena D. Hwang, Maxwell Forbes, and Yejin Choi. 2021. Moral stories: Situated reasoning about norms, intents, actions, and their consequences. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 698–718, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Maxwell Forbes, Jena D Hwang, Vered Shwartz, Maarten Sap, and Yejin Choi. 2020. Social chemistry 101: Learning to reason about social and moral norms. In *EMNLP*.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. Allennlp: A deep semantic natural language processing platform. *arXiv preprint arXiv:1803.07640*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

Dan Hendrycks, Collin Burns, Steven Basart, Andrew Critch, Jerry Li, Dawn Song, and Jacob Steinhardt. 2021. Aligning {ai} with shared human values. In *International Conference on Learning Representations*.

Liwei Jiang, Jena D. Hwang, Chandra Bhagavatula, Ronan Le Bras, Jenny Liang, Jesse Dodge, Keisuke Sakaguchi, Maxwell Forbes, Jon Borchardt, Saadia Gabriel, Yulia Tsvetkov, Oren Etzioni, Maarten Sap, Regina Rini, and Yejin Choi. 2022. Can machines learn morality? the delphi experiment. *arXiv preprint arXiv:2110.07574*.

Vaibhav Kumar and Alan W Black. 2020. Clarq: A large-scale and diverse dataset for clarification question generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7296–7301.

Joseph Kwon, Josh Tenenbaum, and Sydney Levine. 2022. Flexibility in moral cognition: When is it okay to break the rules? In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 44.

Sydney Levine, Max Kleiman-Weiner, Laura Schulz, Joshua Tenenbaum, and Fiery Cushman. 2020. The logic of universalization guides moral judgment. *Proceedings of the National Academy of Sciences*, 117(42):26158–26169.

Chin-Yew Lin and Eduard Hovy. 2002. Manual and automatic evaluation of summaries. In *Proceedings of the ACL-02 Workshop on Automatic Summarization*, pages 45–51.

Alisa Liu, Swabha Swayamdipta, Noah A. Smith, and Yejin Choi. 2022a. WANLI: Worker and ai collaboration for natural language inference dataset creation. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6826–6847, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jiacheng Liu, Skyler Hallinan, Ximing Lu, Pengfei He, Sean Welleck, Hannaneh Hajishirzi, and Yejin Choi. 2022b. Rainier: Reinforced knowledge introspector for commonsense question answering. *arXiv preprint arXiv:2210.03078*.

Nicholas Lourie, Ronan Le Bras, and Yejin Choi. 2021. Scruples: A corpus of community ethical judgments on 32, 000 real-life anecdotes. In *AAAI*.

Aman Madaan, Niket Tandon, Dheeraj Rajagopal, Peter Clark, Yiming Yang, and Eduard Hovy. 2021. Think about it! improving defeasible reasoning by first modeling the question scenario. *arXiv preprint arXiv:2110.12349*.

Bodhisattwa Prasad Majumder, Sudha Rao, Michel Galley, and Julian J McAuley. 2021. Ask what's missing and what's useful: Improving clarification question generation using global knowledge. In *NAACL-HLT*.

Shashi Narayan, Joshua Maynez, Reinald Kim Amplayo, Kuzman Ganchev, Annie Louis, Fantine Huot, Dipanjan Das, and Mirella Lapata. 2022. Conditional generation with a question-answering blueprint. *arXiv preprint arXiv:2207.00397*.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Ratish Puduppully, Yao Fu, and Mirella Lapata. 2022. Data-to-text generation with variational sequential planning. *Transactions of the Association for Computational Linguistics*, 10:697–715.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67.

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.

Rajkumar Ramamurthy, Prithviraj Ammanabrolu, Kianté Brantley, Jack Hessel, Rafet Sifa, Christian Bauckhage, Hannaneh Hajishirzi, and Yejin Choi. 2022. Is reinforcement learning (not) for natural language processing?: Benchmarks, baselines, and building blocks for natural language policy optimization. *arXiv preprint arXiv:2210.01241*.

Sudha Rao and Hal Daumé III. 2018. Learning to ask good questions: Ranking clarification questions using neural expected value of perfect information. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2737–2746.

Sudha Rao and Hal Daumé III. 2019. Answer-based adversarial training for generating clarification questions. 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 143–155.

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, Saadia Gabriel, Lianhui Qin, Dan Jurafsky, Noah A Smith, and Yejin Choi. 2020. Social bias frames: Reasoning about social and power implications of language. In *ACL*.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Zeerak Talat, Hagen Blix, Josef Valvoda, Maya Indira Ganesh, Ryan Cotterell, and Adina Williams. 2022. On the machine learning of ethical judgments from natural language. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 769–779, Seattle, United States. Association for Computational Linguistics.

Alex Warstadt and Omar Agha. to appear. Testing bayesian measures of relevance in discourse. In *Proceedings of Sinn und Bedeutung*, volume 26.

Julia White, Gabriel Poesia, Robert Hawkins, Dorsa Sadigh, and Noah Goodman. 2021. Open-domain clarification question generation without question examples. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 563–570.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

# A Appendix

## A.1 Crowdsourcing and Annotation

Annotations are collected on Amazon Mechanical Turk (MTurk). We run two varieties of HITs: (1) data collection HIT in which we collect questions given statements, and (2) evaluation HIT in which the workers are asked to judge validity of the generated questions. The group of 145 Turkers working on the HIts were manually vetted and selected through an open paid qualification round. We maintain an average pay rate of $15/hour for all HITs.

**Question Collection:** We crowdsource clarification question by prompting annotators with a situation. The crowdworkers are asked to imagine a hypothetical situation where a colleague came to them requesting advice or judgment on the shown situation. The workers are then instructed to write a clarification question they would want to ask that would help them make a better judgment or give a better advice that they would without it. Each of the 6425 situation is presented to 5 distinct annotators; we collect 5 questions per situation. A screenshot of the HIT is shown in Figure 6.

**Human Evaluation:** We ask crowdworkers to evaluate model outputs. Given a situation and a question Turkers are asked to rate the question along three different attributes:

> **Grammaticality** Is the question grammatical? *- yes/no*
>
> **Relevance** Does the question fit the situation and is it plausible that someone might ask this question? - *very relevant/somewhat relevant/entirely irrelevant*
>
> **Informativeness** Can the question lead to new information or does it ask about things

already mentioned in the situation? - *very/somewhat/uninformative*

A screenshot for the evaluation HIT w.r.t grammaticality, relevance, and informativeness is found in Figures 8. Additionally, we evaluate the **defeasibility** of a question. A screenshot of the defeasibility evaluation is shown in Figure 10.

**IRB approval:** We sought and received exemption from our internal IRB. In accordance to the regulations, we do not collect sensitive information. If we do publish WorkerIDs, we will do so by fully anonymizing the information. The exemption received does not require a consent form.

**Language and Demographics:** We have not collected any demographic information from the workers. However, all crowdsourcing was conducted in English and the region (current location of the crowdworker) was set to US. Consequently, what counts as a context of consequence is centered around western views, or views of the English speaking cultures within the United States.

### A.2 Prompting for Answer Generation

One way to elicit a set of opposing answers is through prompting. We instruct GPT-3 to provide a so-called “bad" and a so-called “good" answer to a question about a situation. For the situation *learning how to take a joke* and the question “What was the joke?”, the two answers could be: “it was a lighthearted joke among friends” and “it was an offensive joke”. In order to determine which of the answers is a weakener and which a strengthener, we compare the difference in Delphi’s judgement for $s$ and $s + a_{good}$ or $s + a_{bad}$.

### A.3 Details of PPO

**Policy loss.** To compute the policy loss, we first define the *truncated estimated advantage function*,

$$\hat{A}_t = \sum_{t'=t}^{T-1} (\gamma\lambda)^{t'-t}\delta_{t'},$$
$$\text{where} \quad \delta_{t'} = r_{t'} + \gamma V_\phi(s_{t'+1}) - V_\phi(s_{t'}),$$

where the value function of a certain state $s_t$ is estimated by the value model $V_\phi(\cdot)$, $r_t$ denotes intermediate reward obtained at time step $t$, and $\gamma$ and $\lambda$ denote reward decay factor. PPO then maximizes the empirical expectation of *clipped surrogate objective* term, defined as

$$\text{cso}(\hat{A}_t, r_t(\theta), \epsilon) = \min\big(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t\big),$$

**Value loss.** The value model is trained with MSE loss with respect to the target value, $V_t^{\text{targ}}$:

$$\mathcal{L}_{\text{Value}}(\phi) = \hat{E}\Big[\big(V(s_t;\phi) - V_t^{\text{targ}}\big)^2\Big],$$

### A.4 Implementation Details

The question generation model $\theta_Q$ is trained with T5-large (Raffel et al., 2020) on $\delta$-CLARIFY.

We train the answer generation model $\theta_A$ (T5-large) on 77k instances from $\delta$-CLARIFY$_{silver}$ (§4) and about 4k instances obtained through prompting GPT-3. For the answer-filtering using NLI we use WaNLI (Liu et al., 2022a) as an off-the-shelf NLI model.

We run the PPO algorithm for 6000 steps, with a batch size of 64, top_p= 0.6 and a temperature of 0.7 for the sampling policy. Training for 6000 step took approximately 2 days on 4 A6000, 48G, GPUs.

Our models are based on T5-large which has 770 million parameters.

For the DELPHI model we use the $t5-large$ variant.

### A.5 Licensing

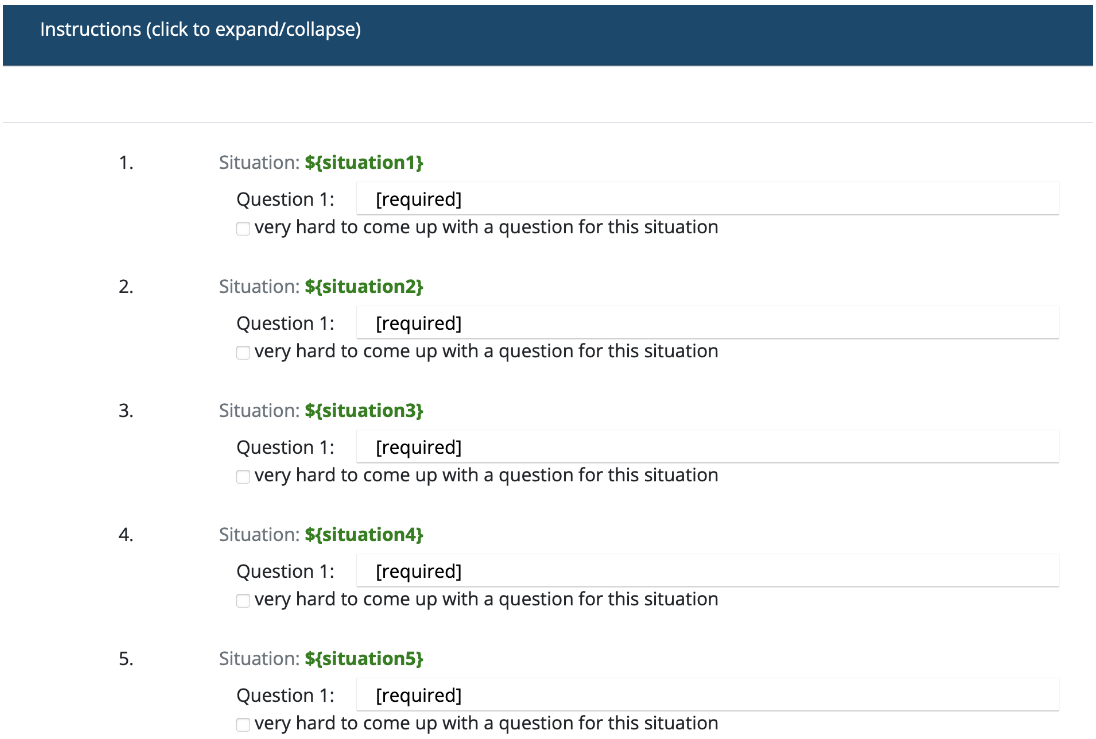
CLARIFYDELPHI and the code will be released under the Apache License 2.0. We will release the dataset under CC-BY.

### A.6 Sample of Generated Questions

We show a sample of 15 generated questions in Tab. 5.

### A.7 Sample of Generated Questions in a Compositional Setting

We show a sample of generated questions in a two-turn interactive setting in Tab. 6.



Instructions (click to expand/collapse)

1. Situation: **${situation1}**

   Question 1: [required]

   □ very hard to come up with a question for this situation

2. Situation: **${situation2}**

   Question 1: [required]

   □ very hard to come up with a question for this situation

3. Situation: **${situation3}**

   Question 1: [required]

   □ very hard to come up with a question for this situation

4. Situation: **${situation4}**

   Question 1: [required]

   □ very hard to come up with a question for this situation

5. Situation: **${situation5}**

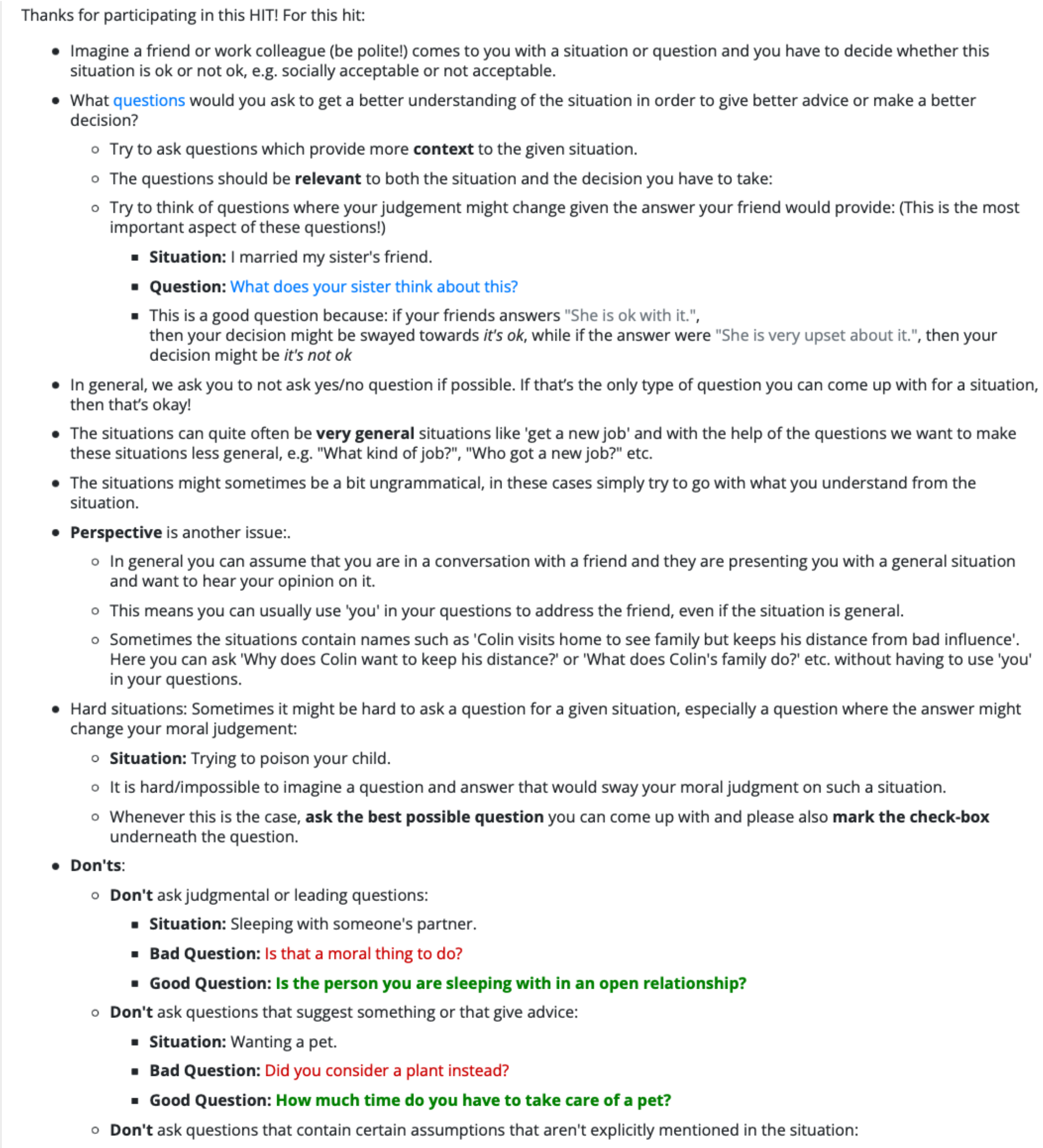
   Question 1: [required]

   □ very hard to come up with a question for this situation

Figure 6: $\delta$-CLARIFY question collection interface.

| Situation | Question |
|---|---|
| sleeping in when you have to get up. | What are you doing when you have to get up? |
| Paying more when you are able to because you make more. | What are you making? |
| Apologizing when you have done something wrong. | What did you do? |
| being careful when opening your car door in a parking lot. | Are you aware of what is inside your car? |
| being out after curfew. | Are you allowed to be out after curfew? |
| respecting your friend's wishes | What is your friend's wish? |
| making a scene | What kind of scene? |
| being friends with benefits with someone | Why are you friends with benefits with them? |
| not liking certain dogs | Why don't you like them? |
| smoking in one's own home | What are the risks? |
| wanting to go home with someone | How do you feel about them? |
| being angry with your partner when they hide things | What are they hiding? |
| Talking about things you know nothing about | Why are you talking about these things? |
| agreeing to go out with someone who bullied you | Why are they bullying you? |
| using your money how you see fit | What are you doing with your money? |

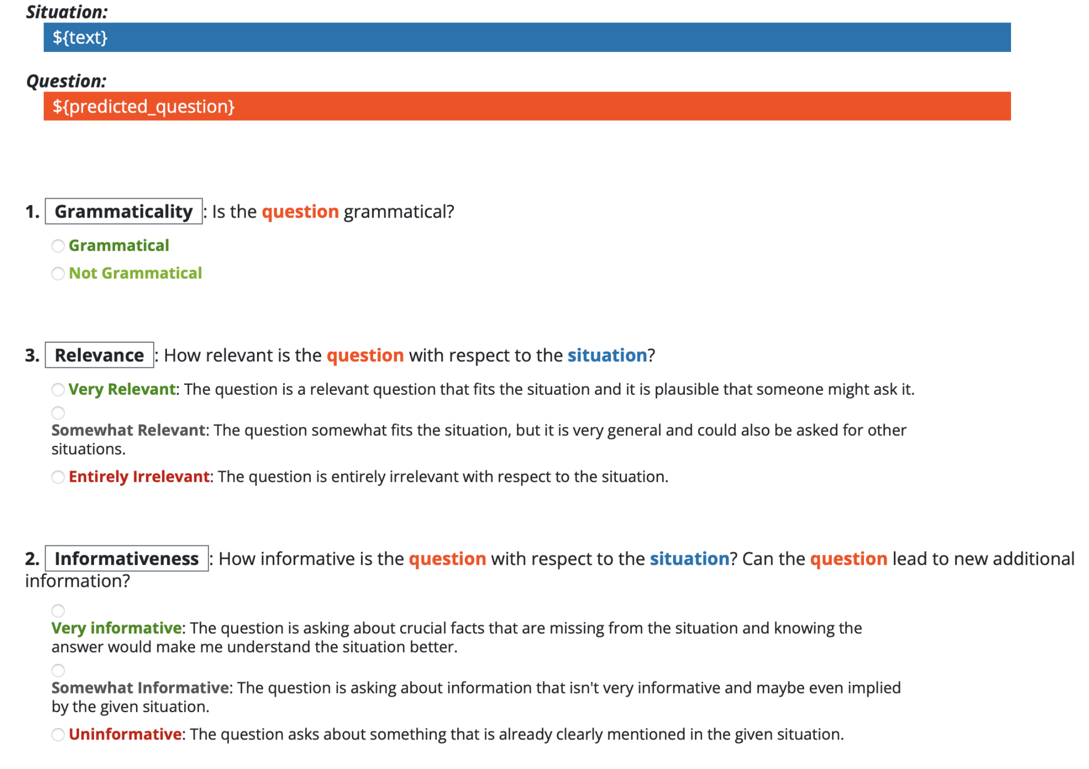
Table 5: Sample of situations from the dev set of SocialChem and the CLARIFYDELPHI generated question.



Thanks for participating in this HIT! For this hit:

- Imagine a friend or work colleague (be polite!) comes to you with a situation or question and you have to decide whether this situation is ok or not ok, e.g. socially acceptable or not acceptable.
- What questions would you ask to get a better understanding of the situation in order to give better advice or make a better decision?
  - Try to ask questions which provide more **context** to the given situation.
  - The questions should be **relevant** to both the situation and the decision you have to take:
  - Try to think of questions where your judgement might change given the answer your friend would provide: (This is the most important aspect of these questions!)
    - **Situation:** I married my sister's friend.
    - **Question:** What does your sister think about this?
    - This is a good question because: if your friends answers "She is ok with it.",<br>then your decision might be swayed towards *it's ok*, while if the answer were "She is very upset about it.", then your decision might be *it's not ok*
- In general, we ask you to not ask yes/no question if possible. If that's the only type of question you can come up with for a situation, then that's okay!
- The situations can quite often be **very general** situations like 'get a new job' and with the help of the questions we want to make these situations less general, e.g. "What kind of job?", "Who got a new job?" etc.
- The situations might sometimes be a bit ungrammatical, in these cases simply try to go with what you understand from the situation.
- **Perspective** is another issue:.
  - In general you can assume that you are in a conversation with a friend and they are presenting you with a general situation and want to hear your opinion on it.
  - This means you can usually use 'you' in your questions to address the friend, even if the situation is general.
  - Sometimes the situations contain names such as 'Colin visits home to see family but keeps his distance from bad influence'. Here you can ask 'Why does Colin want to keep his distance?' or 'What does Colin's family do?' etc. without having to use 'you' in your questions.
- Hard situations: Sometimes it might be hard to ask a question for a given situation, especially a question where the answer might change your moral judgement:
  - **Situation:** Trying to poison your child.
  - It is hard/impossible to imagine a question and answer that would sway your moral judgment on such a situation.
  - Whenever this is the case, **ask the best possible question** you can come up with and please also **mark the check-box** underneath the question.
- **Don'ts**:
  - **Don't** ask judgmental or leading questions:
    - **Situation:** Sleeping with someone's partner.
    - **Bad Question:** Is that a moral thing to do?
    - **Good Question: Is the person you are sleeping with in an open relationship?**
  - **Don't** ask questions that suggest something or that give advice:
    - **Situation:** Wanting a pet.
    - **Bad Question:** Did you consider a plant instead?
    - **Good Question: How much time do you have to take care of a pet?**
  - **Don't** ask questions that contain certain assumptions that aren't explicitly mentioned in the situation:

Figure 7: $\delta$-CLARIFY question collection instructions.



***Situation:***

${text}

***Question:***

${predicted_question}

**1.** **Grammaticality** : Is the **question** grammatical?

○ **Grammatical**

○ **Not Grammatical**

**3.** **Relevance** : How relevant is the **question** with respect to the **situation**?

○ **Very Relevant**: The question is a relevant question that fits the situation and it is plausible that someone might ask it.

○ **Somewhat Relevant**: The question somewhat fits the situation, but it is very general and could also be asked for other situations.

○ **Entirely Irrelevant**: The question is entirely irrelevant with respect to the situation.

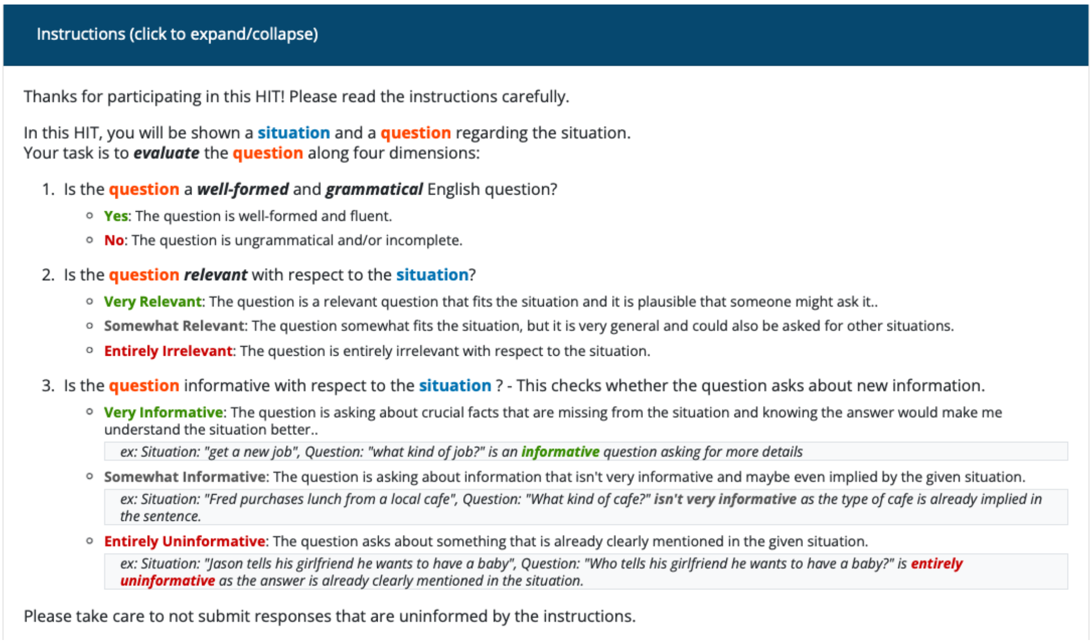
**2.** **Informativeness** : How informative is the **question** with respect to the **situation**? Can the **question** lead to new additional information?

○ **Very informative**: The question is asking about crucial facts that are missing from the situation and knowing the answer would make me understand the situation better.

○ **Somewhat Informative**: The question is asking about information that isn't very informative and maybe even implied by the given situation.

○ **Uninformative**: The question asks about something that is already clearly mentioned in the given situation.

Figure 8: Informativeness, relevance and grammaticality evaluation interface.



Instructions (click to expand/collapse)

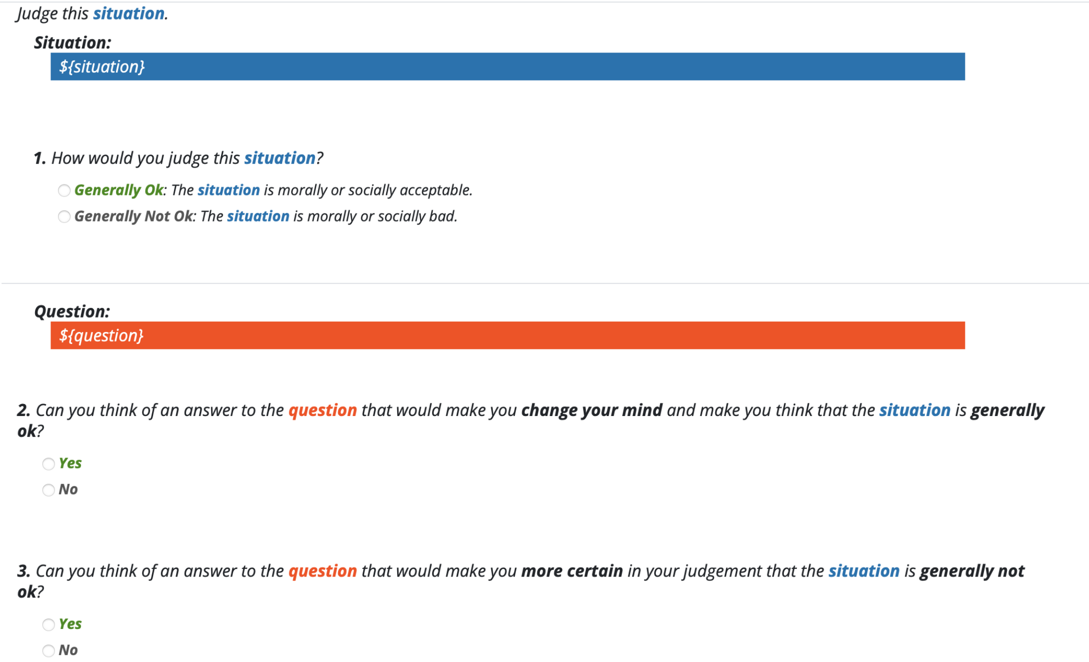
Thanks for participating in this HIT! Please read the instructions carefully.

In this HIT, you will be shown a **situation** and a **question** regarding the situation.
Your task is to ***evaluate*** the **question** along four dimensions:

1. Is the **question** a ***well-formed*** and ***grammatical*** English question?
   - **Yes**: The question is well-formed and fluent.
   - **No**: The question is ungrammatical and/or incomplete.
2. Is the **question** ***relevant*** with respect to the **situation**?
   - **Very Relevant**: The question is a relevant question that fits the situation and it is plausible that someone might ask it..
   - **Somewhat Relevant**: The question somewhat fits the situation, but it is very general and could also be asked for other situations.
   - **Entirely Irrelevant**: The question is entirely irrelevant with respect to the situation.
3. Is the **question** informative with respect to the **situation** ? - This checks whether the question asks about new information.
   - **Very Informative**: The question is asking about crucial facts that are missing from the situation and knowing the answer would make me understand the situation better..
     *ex: Situation: "get a new job", Question: "what kind of job?" is an **informative** question asking for more details*
   - **Somewhat Informative**: The question is asking about information that isn't very informative and maybe even implied by the given situation.
     *ex: Situation: "Fred purchases lunch from a local cafe", Question: "What kind of cafe?" **isn't very informative** as the type of cafe is already implied in the sentence.*
   - **Entirely Uninformative**: The question asks about something that is already clearly mentioned in the given situation.
     *ex: Situation: "Jason tells his girlfriend he wants to have a baby", Question: "Who tells his girlfriend he wants to have a baby?" is **entirely uninformative** as the answer is already clearly mentioned in the situation.*

Please take care to not submit responses that are uninformed by the instructions.

Figure 9: Informativeness, relevance and grammaticality evaluation instructions.



*Judge this **situation**.*

***Situation:***

*${situation}*

***1.** How would you judge this **situation**?*

○ ***Generally Ok**: The **situation** is morally or socially acceptable.*

○ ***Generally Not Ok:** The **situation** is morally or socially bad.*

***Question:***

*${question}*

***2.** Can you think of an answer to the **question** that would make you **change your mind** and make you think that the **situation** is **generally ok**?*

○ ***Yes***

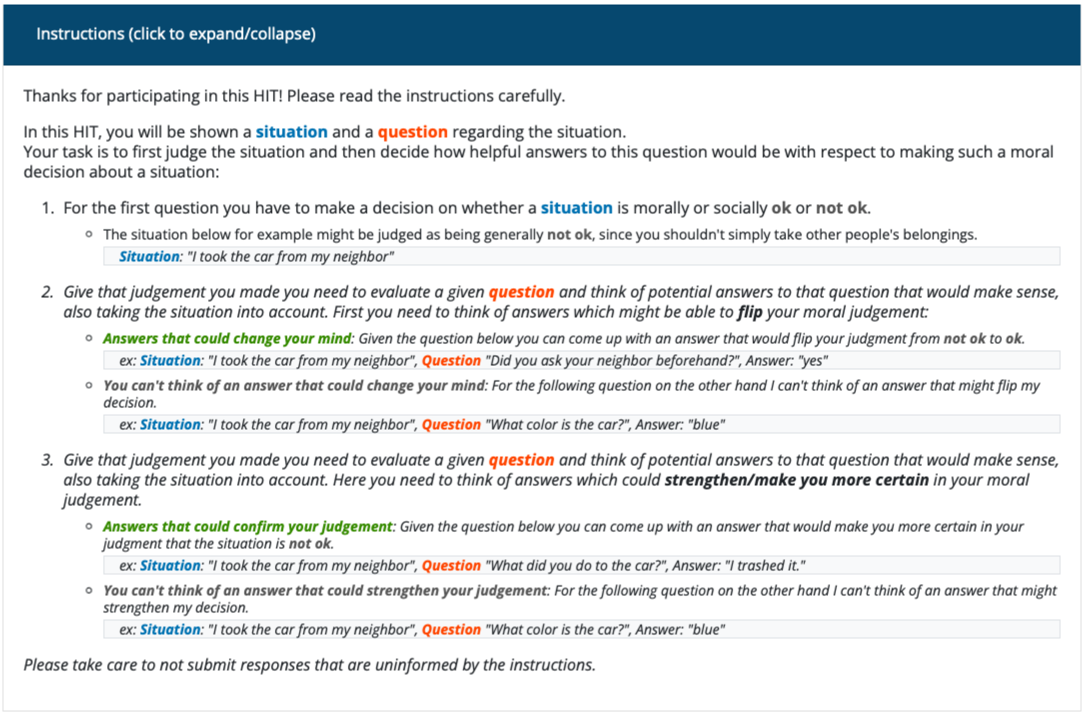
○ ***No***

***3.** Can you think of an answer to the **question** that would make you **more certain** in your judgement that the **situation** is **generally not ok**?*

○ ***Yes***

○ ***No***

Figure 10: Defeasibility evaluation interface.



Instructions (click to expand/collapse)

Thanks for participating in this HIT! Please read the instructions carefully.

In this HIT, you will be shown a **situation** and a **question** regarding the situation.
Your task is to first judge the situation and then decide how helpful answers to this question would be with respect to making such a moral decision about a situation:

1. For the first question you have to make a decision on whether a **situation** is morally or socially **ok** or **not ok**.
   - The situation below for example might be judged as being generally **not ok**, since you shouldn't simply take other people's belongings.
     ***Situation**: "I took the car from my neighbor"*
2. *Give that judgement you made you need to evaluate a given **question** and think of potential answers to that question that would make sense, also taking the situation into account. First you need to think of answers which might be able to **flip** your moral judgement:*
   - ***Answers that could change your mind**: Given the question below you can come up with an answer that would flip your judgment from **not ok** to **ok**.*
     *ex: **Situation**: "I took the car from my neighbor", **Question** "Did you ask your neighbor beforehand?", Answer: "yes"*
   - ***You can't think of an answer that could change your mind**: For the following question on the other hand I can't think of an answer that might flip my decision.*
     *ex: **Situation**: "I took the car from my neighbor", **Question** "What color is the car?", Answer: "blue"*
3. *Give that judgement you made you need to evaluate a given **question** and think of potential answers to that question that would make sense, also taking the situation into account. Here you need to think of answers which could **strengthen/make you more certain** in your moral judgement.*
   - ***Answers that could confirm your judgement**: Given the question below you can come up with an answer that would make you more certain in your judgment that the situation is **not ok**.*
     *ex: **Situation**: "I took the car from my neighbor", **Question** "What did you do to the car?", Answer: "I trashed it."*
   - ***You can't think of an answer that could strengthen your judgement**: For the following question on the other hand I can't think of an answer that might strengthen my decision.*
     *ex: **Situation**: "I took the car from my neighbor", **Question** "What color is the car?", Answer: "blue"*

*Please take care to not submit responses that are uninformed by the instructions.*

Figure 11: Defeasibility evaluation instructions.

| Situation | Question |
|---|---|
| daydreaming sometimes | What are you daydreaming about? |
| daydreaming sometimes about a vacation | What kind of vacation are you planning? |
| stealing money from your parents | How did you steal the money? |
| stealing money from your parents<br>by stealing it from their wallet | How much money did you steal? |
| getting some help when you're lost in life | Are you looking for help? |
| getting some help when you're lost in life and<br>stating that you are not looking for help | Are you having difficulty coping? |
| loving your friend platonically | What is your friend like? |
| loving your friend platonically<br>because he is very nice | What do you like about your friend? |
| being judgmental | How are you being judgmental? |
| being judgmental because<br>I am silently judging them | What is the other person doing? |
| getting cold feet before a wedding | Why are you getting cold feet? |
| getting cold feet before a wedding<br>because you only know your partner for 1 month | What is your relationship status? |
| Being mindful of your health | How are you being mindful? |
| being mindful of your health by working out | What are the benefits of working out? |

Table 6: Sample of situations from the dev set of SocialChem, their expansion through interaction (2 turns) and the CLARIFYDELPHI generated question.

## ACL 2023 Responsible NLP Checklist

### A For every submission:

☑ A1. Did you describe the limitations of your work?
*Limitations and Ethics Statement*

☑ A2. Did you discuss any potential risks of your work?
*Limitations and Ethics Statement*

☑ A3. Do the abstract and introduction summarize the paper's main claims?
*Section 0 and Section 1*

☒ A4. Have you used AI writing assistants when working on this paper?
*Left blank.*

### B ☑ Did you use or create scientific artifacts?

*Section 4*

☑ B1. Did you cite the creators of artifacts you used?
*Section 4*

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
*Appendix*

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
*Appendix*

☑ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
*Appendix and Ethics Statement*

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
*Appendix*

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
*Section 4*

### C ☑ Did you run computational experiments?

*Section 3 and 5*

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
*Appendix*

*The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.*

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
*Appendix*

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
*Section 6*

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
*Appendix*

**D ☑ Did you use human annotators (e.g., crowdworkers) or research with human participants?**
*Section 4*

☑ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
*Appendix*

☑ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
*Appendix*

☑ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
*Appendix*

☑ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
*Appendix*

☑ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
*We only report basic geographic characteristics in the appendix.*