MODELING FUTURE CONVERSATION TURNS TO TEACH LLMS TO ASK CLARIFYING QUESTIONS

Michael J.Q. Zhang[♥,♦], **W. Bradley Knox**[♦], **Eunsol Choi**[♥,♦]
New York University[♥], The University of Texas at Austin[♦]
michaelzhang@nyu.edu

ABSTRACT

Large language models (LLMs) must often respond to highly ambiguous user requests. In such cases, the LLM's best response may be to ask a clarifying question to elicit more information. Existing LLMs often respond by presupposing a single interpretation of such ambiguous requests, frustrating users who intended a different interpretation. We speculate this is caused by current preference data labeling practice, where LLM responses are evaluated only on their prior contexts. To address this, we assign preference labels by simulating their expected outcomes in future turns. This allows LLMs to learn to ask clarifying questions when it can generate responses that are tailored to each user interpretation in future turns. On open-domain QA datasets with multiple annotations, we evaluate systems based on their ability to ask clarifying questions to recover each user's interpretation and expected answer. We compare systems trained using our proposed preference labeling methods against standard methods, which assign preferences based on only prior context. Our method achieves a 5% improvement in F1 measured against the answer set from different interpretations of each query, showing the value of modeling future conversation turns. We further demonstrate that our method can be used to train models to judiciously determine when to ask clarifying questions, directly answering the question when clarification is unnecessary. In our experiments, we find that our method achives a 3% improvement in accuracy of such judgments over existing methods.

1 INTRODUCTION

Ambiguity is a hallmark of natural language that enables concise communication by allowing speakers to exclude details that are inferable from the context (e.g., conversational, temporal, geographical) (Piantadosi et al., 2012). At times, however, the speaker's intent is unclear despite the context, and further interaction is necessary to clarify their intent. Asking clarifying questions is particularly important for large language models (LLMs), which are tasked with serving a wide audience, often without access to the personalized context available in human interactions. In this work, we develop LLMs that can ask clarifying questions to resolve ambiguity in their users' requests.

State-of-the-art LLMs (OpenAI, 2023; Gemini Team, 2024) often do not ask clarifying questions when presented with an ambiguous request, and instead respond directly by assuming the user's intent (see an example in Figure 1). We speculate that this tendency stems from a shortcoming in their RLHF training pipelines, which utilize annotated preference data (Ouyang et al., 2022). In standard preference data collection, annotators are given a conversation history and are tasked with ranking options for the next assistant turn (Bai et al., 2022; Wang et al., 2024). These annotation schemes only consider preferences over single-turns of interaction, making it difficult for annotators to assess the utility of a clarifying question. Furthermore, this can lead to biases where annotators prefer responses with complete but presumptuous answers over incomplete clarifying questions.

We propose an alternative method for annotating *double-turn* preferences over clarifying questions where annotators interact with the LLM by providing their responses to clarifying questions and observing the LLM's subsequent responses. Each annotator then assigns preferences based on whether the completed interaction successfully fulfilled the request. When comparing multiple clarifying

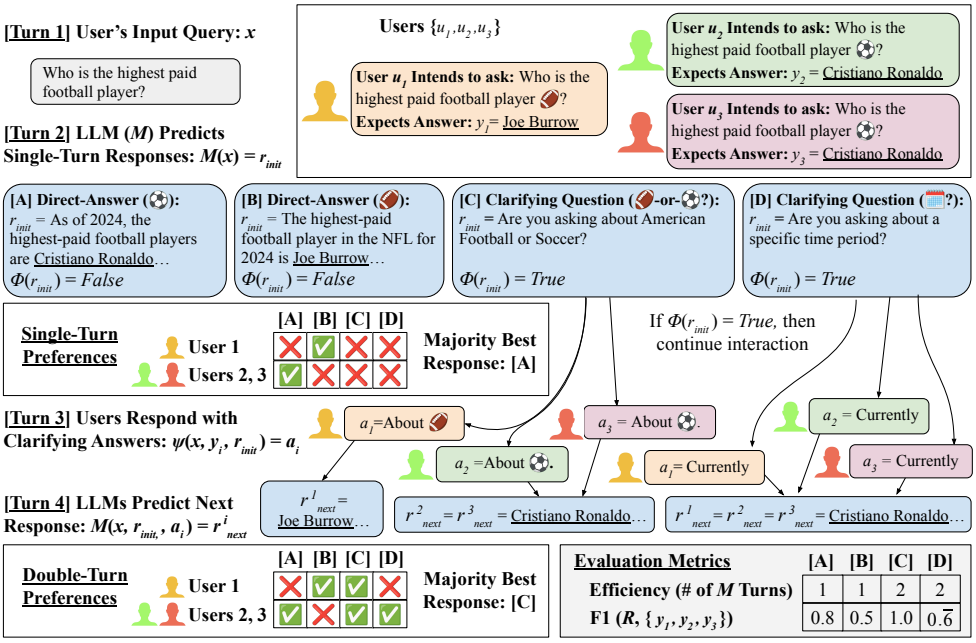

Figure 1: Depiction of our interaction scenario and preference labeling schemes. We aim to build an LLM that can interact with user to generate the final answer set R, containing an answer for each user, for the input query x. In this example, we include two responses from state-of-the-art LLMs ([A] from GPT-4 and [B] from Gemini, full responses in Appendix A), which both presuppose an interpretation of the word football. We also include two clarifying responses ([C] and [D]) where [C] correctly disambiguates the two intended interpretations across all users. We depict two ways to assign preferences on LLM's initial output, single-turn and our proposed double-turn.

questions and direct answer responses, each annotator's preferences are aggregated to identify the response that maximizes preference across all annotators. We depict our proposed double-turn preference annotation method and compare against standard methods in Figure 1.

To demonstrate the benefits of training LLMs to ask clarifying questions with double-turn preferences, we experiment on open-domain QA (Chen et al., 2017). We develop an automatic evaluation framework for evaluating clarifying questions that use simulated user interactions (Section 2). We evaluate systems on two axes: *efficiency* measured by the number of model turns and *effectiveness* measured by F1 score between the predicted answer set for each user and their expected gold answer.

Our experimental results show that training systems with double-turn preferences outperforms training with standard preferences annotation methods (Zhu et al., 2023), resulting in consistent 4-5% improvement in F1 score over three base LLMs (Dubey et al., 2024; Mesnard et al., 2024). We also demonstrate that double-turn preferences can be used to train systems that determine if clarification is needed, or if the user's question can be answered without additional interaction.

We summarize our contributions below:

- We establish an automatic framework for evaluating double-turn interaction involving clarifying question. Our framework includes a user simulation model and automatic evaluation metrics measuring the system's ability to predict outputs that are tailored to each user efficiently.
- We develop a *double-turn* preference annotation method for labeling preferences over clarifying question and direct response that utilizes the outcomes in later turns of interaction.
- Our results on Open-Domain QA datasets (Kwiatkowski et al., 2019; Min et al., 2020) demonstrate that training systems to ask clarifying questions using our double-turn preferences improves QA performance over training with standard single-turn preferences. We also show that training with *double-turn* preferences can teach models to judiciously decide when to ask for clarification.

Our evaluation paradigm and preference labeling scheme can further support future research in developing interactive LLM assistants. We release all code and data at `https://github.com/mikejqzhang/clarifying_questions`.

2 TASK: QA WITH SIMULATED USER INTERACTION

We chose to study open-domain QA, as it is both prevalent in real user-assistant interactions (Zhao et al., 2024) and the scope of necessary clarification questions is not immediately identifiable from the input or the task. For simplicity, we choose a well-studied short factoid QA setting (Kwiatkowski et al., 2019; Min et al., 2020) where an answer to the question consists of a few tokens. Below, we describe our task and evaluation metrics. Figure 1 contains an example of task scenario.

2.1 INTERACTION SCENARIO

Setting / Goal: We assume an input query x, which can be potentially ambiguous, and a set of k users $u_1, ...u_k$ who posed the input query x. Each i-th user u_i has an disambiguated intent for query (x_i) and an expected gold answer (y_i). Our goal is to build a system M that returns an answer set R that can satisfy as many users as possible (by providing the expected answer y_i for each user u_i) with the minimum number of interactions (measured by the number of model turns).

We will have a simple function $\phi(M(x))$ that returns *True* if $M(x)$ is a clarifying question, and *False* otherwise. We further assume an oracle user simulation model $\psi(x, r_{init}, y_i) = a_i$, that takes first turn input x, model's initial output $M(x)$, and i-th user's targeted answer y_i, that generates a response to the model's initial response $M(x)$.

Dialogue Turns: In the initial turn, user poses the input query x. In the second turn, given the input query x, the system M is to generate its initial response (r_{init}). Depending on whether r_{init} is a question, it will trigger different dialogue trajectories.

- **Double-turn Scenario** If the initial response is a clarifying question ($\phi(r_{\text{init}})$=*True*), we continue the conversation with the user simulation model. For each of i-th user, the evaluation system generates a third turn (a_i) with the user simulation model $\psi(x, r_{\text{init}}, y_i) = a_i$. For the fourth turn, the system will take each of the k interaction traces $(x, r_{\text{init}}, a_i)$ to generate the fourth turn response for i-th user $M(x, r_{\text{init}}, a_i) = r^i_{\text{next}}$. The final output answer set from the system will be a set of k answers $R = \{r^1_{\text{next}}, r^2_{\text{next}}, ...r^k_{\text{next}}\}$, where each r^i_{next} is a response for i-th user.
- **Single-turn Scenario** Otherwise if $\phi(r_{\text{init}})$=*False*, we will consider the model output $R = \{r_{\text{init}}\}$ as the final prediction. To allow a model that does not generate clarifying questions to generate multiple answers, we sample multiple answers from the model, up to k answers. The final answer set in this case will be $R = \{r^1_{\text{init}}, ...r^k_{\text{init}}\}$.

Existing QA models mostly generates an single answer to the initial input query x, even when the query is ambiguous. Such systems that predict an answer without first confirming the user's intent may mislead users with incorrect answers and fail to serve the entirety of their user population. In this work, we explore methods for *QA-with-Clarification*, where models can ask the user a clarifying question q and observe the user's response a before predicting an answer. A system can also sometimes poses a clarifying question and sometimes directly answer (Zhang & Choi, 2023a).

Evaluations: We evaluate the system in two axis: efficiency and effectiveness. To measure efficiency, we simply measure the average number of total turns from the system. For unambiguous inputs, asking clarifying question is unnecessary. If the system can satisfy users without incurring additional interaction, such system should be preferred.

To measure effectiveness, we use **F1** between the model generated answer set (R) and the gold reference answer set ($\{y_1, ..y_k\}$). When evaluating QA-with-Clarification systems, we additionally enforce that answers must be correct for their corresponding clarifying answer. We hold QA-with-Clarifications to this higher standard, as it aligns with our goal of to developing systems that can identify different user interpretations of a query and accurately predict corresponding answers.

2.2 EVALUATION FRAMEWORK IMPLEMENTATION

Data: We will leverage existing open-domain QA datasets (Kwiatkowski et al., 2019; Min et al., 2020) where each query is paired with annotated answers from multiple annotators.

Identifying clarifying question $\phi(\cdot)$: We will use a very simple method, which takes the input string and check whether it starts with "Clarifying Question:".

User Simulation Model ψ for Turn 3: We use GPT-4 as our oracle user simulation model (ψ). We prompt GPT-4 with the input query x, clarifying question q, and target answer y_i and task it with generating a corresponding clarifying answer ($(x, q, y_i) \rightarrow a_i$). We prompt GPT-4 to abstain from providing a clarifying answer a_i if it judges that none exists. This happens when the clarifying question does not address the ambiguity in the query, hence we count the resulting target answer prediction r^i_{next} as incorrect. We further ensure that the generated clarifying answers do not leak the target answer by removing instances where the target answer y_i appears in the predicted clarifying answer a_i, treating such examples the same as abstains from GPT-4 (full prompt in Appendix B).

3 Fine-tuning LLMs to ask Clarifying Questions

We present our approach to build a model that can engage with users for a multi-turn interaction, ask clarifying question when it can satisfy more users in the future turns. Following the standard RLHF training pipeline (Ouyang et al., 2022), we first construct an instruction tuning dataset focusing on clarifying responses as desired outputs. We then use this constructed dataset for supervised finetuning (SFT), before performing preference learning between two possible responses. We first describe our method for constructing this instruction tuning dataset before introducing our annotation scheme for assigning preferences over clarifying questions.

3.1 Preference Data Generation from User-LLM Interactions

Our method for labeling preferences deviates from standard annotation methods (Bai et al., 2022; Wang et al., 2024) in two ways. First, we simulate an additional interaction with the user and derive reward from the LLM's final response after the additional turn. Second, we expect multiple annotators for each example. By using multiple human annotators to identify a set of expected answers, we simulate each annotator's interactions with the LLM and their preferences. While prior work has explored using multiple annotator judgments in single-turn preference datasets (Wang et al., 2024; Kopf et al., 2023), these works aggregate annotator judgments via majority choice to generate a single preference label per example. This practice, however, can lead to adverse outcomes where systems whose outputs are catered toward a single, majority accepted response (Fleisig et al., 2023; Santy et al., 2023). In contrast, we use preferences across multiple annotators without removing individual annotator judgments, helping us identify whether a clarifying question or direct-answer response successfully accommodates the range of different annotator's interpretations.

In our annotation scheme (depicted in Figure 2), annotators are provided an input query x with several candidate clarifying question q and direct-answer responses $\widehat{y}$. They respond to each clarifying question by providing the clarifying answer that corresponds to their interpretation. We then show annotators the model's final output prediction, given each clarification. Annotators then assign their preferences over clarifying questions and direct-answer responses based on whether the final output matched their expected interpretation. We then aggregate these preferences to identify which response was preferred by the most annotators.

To prevent systems from asking unnecessary clarifying questions, ties between direct-answer and clarifying question responses are broken by favoring the direct-answer response. The final aggregated preferences can then be used in conjunction with standard RL methods from human preferences for training. While this annotation scheme can be applied to human annotators, in this work we use simulated user interactions for annotation, simulating different users' interactions from existing datasets with multiple annotations (discussed below in Section 4).

Simulating User Interaction To simulate different user interpretations of the input query, we utilize datasets consisting of an input query x and a set of correct answer outputs $\{y_1, ..., y_k\}$. For each query, we use the set of gold outputs to simulate k different users, one for each answer in $\{a_1, ..., a_k\}$ which are identified by multiple human annotators (we discuss how these annotations are sourced in Section 4). We then simulate the behaviors of each of these users and their responses to a proposed clarifying question q using a trained *user-simulator* model, which takes as input query x, clarifying question q, and the user's annotated answer y_i and is trained to predict the user's clarifying answer $(x, q, y_i) \rightarrow a_i$. While we use prompted GPT-4 to simulate clarifying answers from users for evaluation, during training we use this trained user-simulator model to reduce costs and to ensure that our systems do not overfit to a particular user-simulator method.

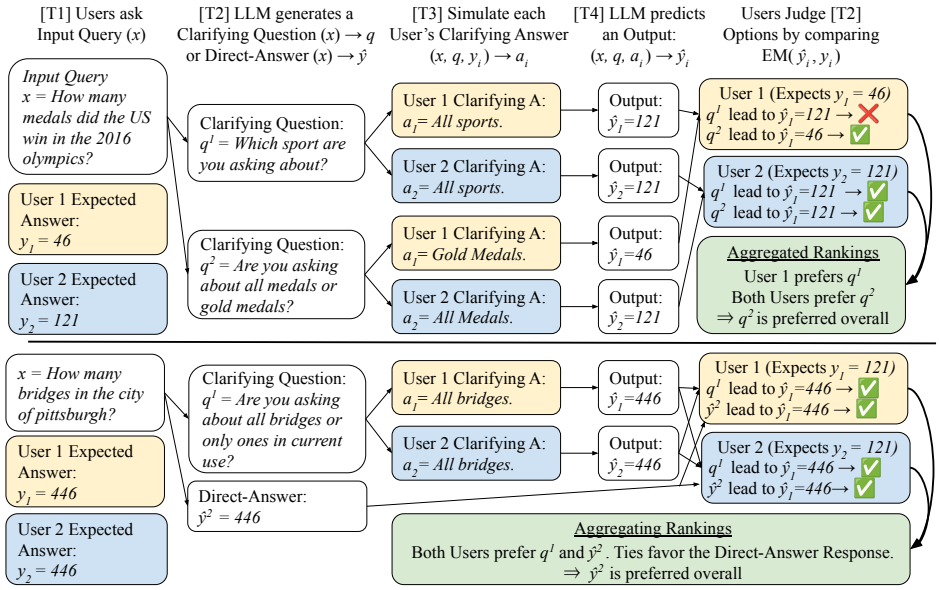

Figure 2: Depiction of our preference annotation method. Here, simulated users provide their responses to model-generated clarifying questions and determine preference based on which clarifying question or direct-answer responses lead to their expected answer. We then aggregate preferences across users by selecting the response that is preferred by the most users while minimizing the number of user interactions turns.

Answer Prediction / Assigning Preferences Conditioning on the simulated clarifying answer, our LLM model predicts the output $\widehat{y_i}$, $(x, q, a_i) \rightarrow \widehat{y_i}$. As a last step, we aggregate annotator preferences so that clarifying questions are ranked by their EM accuracy, evaluated against each annotator's expected answer: $\sum_{i=1}^{k} \text{EM}(\widehat{y_i}, y_i)/k$. To rank direct-answer responses, which only predict a single answer $\widehat{y}$, we similarly evaluate the EM accuracy of the predicted answer $\widehat{y}$ against each annotator's expected answer. To prevent systems from asking unnecessary clarifying questions, we favor the direct-answer response in the case of ties. We then use these aggregated rankings to determine the pairwise preference labels between responses for training.

4 Experimental Settings

Models We use Llama2-7b (Touvron et al., 2023), Gemma-7b (Mesnard et al., 2024), and Llama3-8b (Dubey et al., 2024) as our base LLMs. We forego using the instruction-tuned models due to data leakage concerns, as such systems are finetuned on the entirety of NQ-Open. For training and inference, we use 8-bit quantization (Dettmers et al., 2022) with LoRA (Hu et al., 2022; Dettmers et al., 2024) (training details in Appendix B).

Data We perform our experiments on the NaturalQuestions (NQ-Open) (Kwiatkowski et al., 2019; Lee et al., 2019) and AmbigQA (Min et al., 2020) datasets. In both of these datasets, each input query x are associated with a set of possible output answers $\{y_1, \ldots, y_k\}$. We describe each dataset below, and provide details in Appendix A.

- **NQ-Open** is comprised of questions from Google search queries by real users. Each query is annotated with answers from Wikipedia by up to five annotators; many questions (about 10% of train and 42% of development set) in the dataset contain multiple distinct answers based on each annotator's judgment in interpreting the query and selecting the best answer span. The answers sets identified by annotators for such examples can stem from a variety of ambiguities, from formatting to ambiguities resulting in semantically distinct answers (examples in Appendix A).
- **AmbigQA** additionally annotates a subset of NQ-Open whether the question is ambiguous. If the input is ambiguous, annotators then provide the additional answers to other possible interpreta-

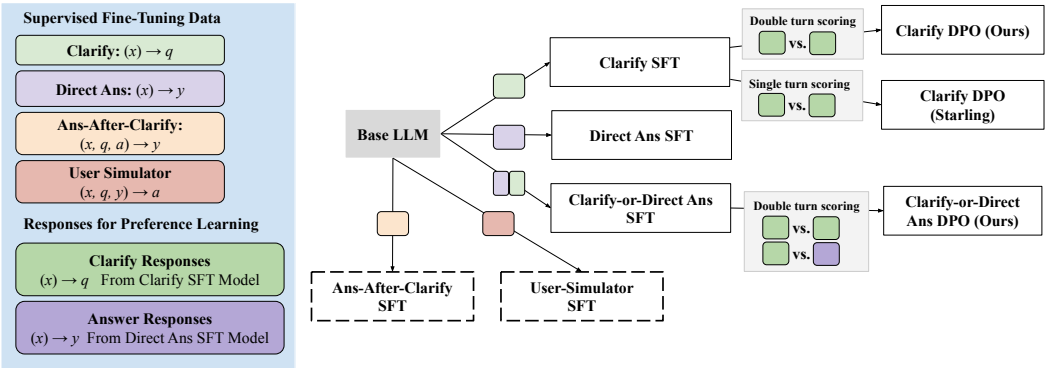

Figure 3: Illustration of models in our study (right) and the data used for training them (left). Ans-After-Clarify SFT model is used to generate responses for the fourth turn, User-simulator SFT model is used to generate responses for the third turn. All other models generate responses for the 2nd turn.

tions of the query. This process identifies about 56% of all queries in NQ-Open as ambiguous, and recovers 2.6x more answers to ambiguous questions.

We evaluate on AmbigQA test set (n=1960). In addition to evaluating over the **Full** test set, we additionally report performance on splits of **Unambiguous** questions (788 questions with one answer each) and **Ambiguous** questions (1,172 questions averaging 3.7 answers each).

Supervised Finetuning (SFT) Dataset Generation To generate examples for SFT training, we prompt GPT-4 with an input query x and a set of feasible answers $Y = \{y_1, \ldots, y_k\}$ and task it with proposing a clarifying question q that will help determine which answer the questioner expects. Our prompt also tasks GPT-4 with predicting responses to the clarifying question for each one of the feasible answers $\{(a_1, y_1), \ldots, (a_k, y_k)\}$, if it determines one exists. We use the same GPT-4 prompt to construct all examples SFT training, and introduce two methods for generating feasible answer sets Y for a given query x.

- **Answers from Human Annotations:** We use the ambiguous queries from the AmbigQA training and development splits and their annotated answers as the feasible answer set Y_{human}. Identifying ambiguities and labeling all possible answers is a challenging task for annotators, and thus this data can be expensive to collect. Furthermore, we hypothesize that what is ambiguous for an LLM often deviates from what is ambiguous for humans.
- **Answers from Model Predictions:** We build feasible answer sets Y_{model} for queries in the NQ-Open training set using answer candidates proposed by a base LLM. Prior work has demonstrated that base LLMs are well calibrated in open-domain QA tasks with few-shot prompting (Kadavath et al., 2022; Cole et al., 2023). Drawing on these observations, we construct a feasible answer set Y_{model} by sampling 5-shot predictions from Llama-2-7b (Touvron et al., 2023). For each query, we sample 5-shot prompt and generate its greedy answer with temperature $T = 0.0$ and sample an answer with $T = 1.0$. We repeat the process 10 times per query and remove repeated answers to generate a set of model-identified feasible answers. We further filter examples where none of the feasible answers match any annotated answers from NQ-Open.

For each method of generating feasible answer sets (Y_{human} and Y_{model}), we generate an SFT dataset of 4,400 input query and clarifying question pairs (x, q), which we split into training (4,000) and development (400) splits. Between both datasets, this gives us a total of 19,807 (x, q, a_i, y_i) examples. In Appendix A, we include examples and the exact prompts.

4.1 Compared Systems

Figure 3 summarizes all models used in our experiments. We have two types of models, model that will be used at the second turn (which we refer as Turn 2 model) and model that will be used at fourth turn (which we refer as Turn 4 model and notate as "Ans-After-Clarify" model in the figure). The model that will be used at the second turn can ask clarification question or directly answer the question. All models that are capable of asking clarification questions are paired with the same

Turn 4 model, which is trained over all generated SFT examples. We use a separate model for performing this latter step to ensure that differences in performance only reflect the value of the clarifying question, and explore training joint models for both steps in Section 5. We describe each of these below.

Direct Ans SFT This model is trained to predict the answer given the input query $(x) \rightarrow y$. We fine-tune base LLM with next-token prediction loss on the full set of question answer pairs (i.e., each (x, y_i)) in our generated SFT training sets. We use this model under two inference settings: **(1) Greedy** where we take the single greedy answer prediction and **(2) Sampled** where we sample a set predicted answers from the LLM. We sample 20 answers from the model and select the top k most frequent answers, where k matches the gold number of annotated answers.[1]

Clarify SFT This model is trained to ask clarifying question to input query. We finetune base LLMs on our SFT datasets constructed with both human annotations and model predicted answers. In Section 5, we provide ablation experiments training over each dataset to compare their utility.

Clarify DPO (Starling, Ours) These models are each fine-tuned from Clarify SFT model with using DPO (Rafailov et al., 2023), a method for optimizing LLMs on preference data. It uses the following loss for updating the policy LLM π_θ based on the base reference policy π_{ref}, prompt h, and preferred/rejected responses (r_p, r_r):

$$L(\pi_\theta, \pi_{ref}, r_p, r_r, h) = \log \sigma(\beta \log \frac{\pi_\theta(r_p|h)}{\pi_{ref}(r_p|h)} - \beta \log \frac{\pi_\theta(r_r|h)}{\pi_{ref}(r_r|h)})$$

where β is a hyperparameter and $\pi(r|h)$ is the predicted probability of generating r given the prompt h. These models are provided the same response pairs sampled from Clarify SFT model,[2] only differing in how reward are assigned.

- Starling. This uses off-the-shelf reward model to rank clarifying questions. We use the Starling-RM-7B-alpha reward model (Zhu et al., 2023) which was finetuned on over 3.8M single-turn preferences between the outputs from a variety of LLMs.
- Ours. We rank clarifying questions by whether the predicted answer **matches** each user's gold answer after observing each simulated user's clarifying answer, averaged over all simulated users.[3]

Clarify-or-Direct Ans DPO (Ours) We further train our Clarify-or-Direct Ans SFT model above using double-turn preferences over clarifying question and direct-answer responses with DPO learning objective. We perform DPO training over pairwise comparisons from the same six candidate clarifying questions generated for our Clarify DPO models above and the single greedy sampled direct-answer response generated from our Direct Ans SFT model above.

5 RESULTS

We report our main results in Table 1. Overall, systems that are equipped to ask clarifying questions outperform Direct-Ans models in effectiveness (Answer F1). Clarify-or-Answer methods strike a balance between effectiveness and efficiency.

Training with our double-turn preference labeling scheme consistently yields the best answer F1 overall. We observe mixed results from training with single-turn RLHF annotations from Starling: yielding positive results with Llama2 as the base LLM and minor increases/decreases in performance for Gemma and Llama3 when evaluated over both ambiguous and unambiguous queries. These results support our claim that single-turn RLHF annotation struggles to identify useful clarifying questions. In contrast, training with double-turn preferences, where clarifying questions are assessed based on their later outcomes, demonstrates consistent improvements in the LLM's ability to generate useful clarifying questions, on both ambiguous and unambiguous queries. This demonstrates

[1] In rare cases (< 2%), there are less than k unique answers within the 20 samples.

[2] We sample six candidate clarifying questions (one with $T = 0.0$ and five with $T = 1.0$) and form preference pairs among them.

[3] Alternative ways of scoring clarifying question is by using the likelihood of gold answer after the answer to the clarifying question, following prior work (Andukuri et al., 2024). We provide results with this setting in Appendix C.

Table 1: Main results, separated by base model and systems that never clarify (Avg Turns = 1), always clarify (Avg Turns = 2), or can do both (Avg Turns $\in [1, 2]$). Our method (Clarify DPO (Ours)) yields the optimal results in Answer F1 for all base models at the cost of always asking clarifying questions. Our method that selectively asks clarifying questions (Clarify-or-Direct-Ans DPO (Ours)) strikes a balance between efficiency (Avg Turns) and effectiveness (Answer F1). Bold results denote statistically significant differences from all other methods with the same base model (over $N = 10,000$ samples with $p < 0.01$).

Base	Turn 2 Model	Avg Turns (↓)	Answer F1 (↑)		
			Unambig	Ambig	Total
Llama2 (7b)	Direct-Ans SFT	1	0.254	0.168	0.211
	Direct-Ans SFT (Sampled)	1	0.250	0.172	0.214
	Clarify SFT	2	0.310	0.216	0.259
	Clarify DPO (Starling)	2	0.310	0.257	0.283
	Clarify DPO (Ours)	2	**0.383**	**0.282**	**0.328**
	Clarify-or-Direct-Ans DPO (Ours)	1.56	0.289	0.211	0.243
Gemma (7b)	Direct-Ans SFT	1	0.261	0.168	0.211
	Direct-Ans SFT (Sampled)	1	0.237	0.179	0.214
	Clarify SFT	2	0.357	0.236	0.288
	Clarify DPO (Starling)	2	0.339	0.257	0.295
	Clarify DPO (Ours)	2	**0.407**	**0.286**	**0.339**
	Clarify-or-Direct-Ans DPO (Ours)	1.61	0.282	0.222	0.246
Llama3 (8b)	Direct-Ans	1	0.312	0.192	0.248
	Direct-Ans (Sampled)	1	0.282	0.202	0.247
	Clarify SFT	2	0.376	0.265	0.315
	Clarify DPO (Starling)	2	0.362	0.267	0.309
	Clarify DPO (Ours)	2	**0.429**	**0.317**	**0.365**
	Clarify-or-Direct-Ans DPO (Ours)	1.57	0.352	0.251	0.291

that clarifying questions not only help models disambiguate user intents for ambiguous queries, but also can help models recover correct answers in general even for unambiguous queries. We include examples of generated clarifying questions in Appendix D. In Appendix C, we further compare our results with adapted approaches from related work (Andukuri et al., 2024; Deng et al., 2023a;b), which underperform our proposed systems.

Ablations: Comparing Clarifying Question Generated with Human Annotated Answers vs. Model Predicted Answers So far, we performed SFT training on the union of clarifying questions generated from model-identified (Y_{model}) and human-identified (Y_{human}) feasible answer sets. Is one of them more useful than the other? In Table 2, we compare performance using SFT examples constructed from our Y_{model} and Y_{human} answer sets, after RLHF training with double-turn preference.

Table 2: Answer F1 results comparing training Clarify DPO (Ours) systems using SFT training on clarifying questions generated from Y_{human} or Y_{model} answer sets or both.

Model	SFT Data	Unambig	Ambig	Full
Llama2 (7b)	Human	0.288	0.236	0.259
	Model	0.299	0.264	0.283
	Both	0.383	0.282	0.328
Gemma (7b)	Human	0.379	0.274	0.323
	Model	0.391	0.283	0.329
	Both	0.407	0.286	0.339
Llama3 (8b)	Human	0.401	0.309	0.349
	Model	0.423	0.312	0.364
	Both	0.429	0.317	0.365

With Llama2 base model, we see clear improvements when using our model-ambiguity SFT dataset over using our human-ambiguity dataset alone. These improvements even hold when evaluated against the answer sets from AmbigQA, despite the fact that our human-ambiguity SFT dataset was generated to distinguish between answers labeled in the AmbigQA training split. However, on Gemma and LLama3, we find only minor improvements from training with model-ambiguity SFT dataset. The model-ambiguity dataset was using feasible answer sets Y_{model} from Llama2 predictions. This gap suggests that constructing a model-ambiguity dataset that is specific to the base model improves the resultant clarifying question quality.

A Joint Model for Asking Clarifying Questions and Predicting Answers Here, we experiment with using a single LLM for both the clarification $(x \rightarrow q)$ and answer $(x, q, a_i \rightarrow y_i)$ conversation turns. To create this joint model, we take inspiration from prior works demonstrating that averaging the weights of multiple finetuned models is a simple and effective approach to joining the capabilities of finetuned models into a single LLM (Wortsman et al., 2022). We merge the LoRA parameter updates from our clarification model (trained with preferences from full interactions) and the answer prediction models used in our main results (Table 1).

Table 3 compares the joint model against using separate LLMs for each conversation turn. Overall performance degrades when using a joint model for both turns; however, we are still able to retain most of the performance gains over Direct-Ans and QA-with-Clarification methods trained with single-turn preferences. Future work may explore alternative training methods that mitigate the performance degradation, such as using mixture-of-experts methods for LLMs (Jiang et al., 2024) or improved multi-turn learning algorithms (Zhou et al., 2024b).

Table 3: Answer F1 results comparing separate versus joint clarifying question and answering models (Cl / Ans).

Model	Cl / Ans	Unambig	Ambig	Full
Llama2 (7b)	Separate	0.383	0.282	0.328
	Joint	0.324	0.250	0.287
Gemma (7b)	Separate	0.407	0.286	0.339
	Joint	0.362	0.274	0.314
Llama3 (8b)	Separate	0.429	0.317	0.365
	Joint	0.410	0.292	0.346

Determining When Clarifying Questions are Necessary We further investigate our system's ability to identify when clarifying questions are necessary. To evaluate our Clarify-or-Answer model's predictions, we report **Direct-Answer Accuracy**: Did the the system correctly decide to directly answer the question when the question was unambiguous and the greedy predicted direct-answer was correct. To determine whether the greedy predicted direct-answer was correct, we decode our Clarify-or-Answer system's greedy direct-answer prediction for each input question. We also report **Ambig Acc**: the accuracy of a system's clarify-vs-answer predictions on the human ambiguous-vs-unambiguous labels.

We compare our models predictions (C-or-A-Pred) against a **random** baseline. To compute the random baseline, we fix the percent of direct-answer responses (DA%) to our model's clarify-or-answer predictions and randomly sample DA% of predictions to directly answer, and predict clarifying questions for the remaining queries. We report our results in Table 4. Our Clarify-or-Answer system's predictions consistently outperforms all baselines across all metrics. The relatively large gains on Direct-Answer accuracy compared to the F1, further indicates that we can expect performance on end-task metrics to further improve with better clarifying question generations.

Table 4: Results for Clarify-or-Answer methods. For each model, we include the percent of questions directly answered (DA%), and evaluate using F1, Direct-Answer (DA) Acc., and Ambig (A) Acc.

Model	Method (DA%)	F1	DA Acc	A Acc
Llama2	Random (44%)	0.234	0.554	0.521
	C-or-A Pred (44%)	**0.243**	**0.619**	0.537
Gemma	Random (38%)	0.237	0.604	0.510
	C-or-A Pred (38%)	**0.246**	**0.643**	**0.543**
Llama3	Random (43%)	0.280	0.549	0.511
	C-or-A Pred (43%)	**0.365**	**0.595**	**0.540**

6 Related Work

Ambiguity in NLP Ambiguity has been studied extensively in NLP in across a variety of tasks. In machine translation, works have studied instances where sentences in a source language have multiple valid translations in a target language (Fernandes et al., 2021; Lopes et al., 2020; Voita et al., 2019). In natural language inference (NLI), Nie et al. (2020) collected a dataset of highly cross-annotated examples that contain high disagreement in annotator judgments. Liu et al. (2023) then found that these disagreements are often the result of ambiguity in the input. Recent work has studied various sources of ambiguity, from entity-linking (Lee et al., 2024), co-reference ambiguities (Yuan et al., 2023), to temporal and geographical contexts (Zhang & Choi, 2021).

Uncertainty in LLMs Several recent works have studied calibration in LLMs, looking specifically at the effects of RLHF training. Specifically, whereas likelihoods for base LLMs (i.e., pretrained) tend to be well-calibrated for a variety of tasks (Cole et al., 2023; Kadavath et al., 2022), this be-

havior is lost after RLHF finetuning (OpenAI, 2023). Additionally, Zhou et al. (2024a) investigated uncertainty-expression via epistemic markers generated by LLMs and found evidence that standard single-turn RLHF training encourages over-confidence in LLM responses. Band et al. (2024) trained an LLM to generate long-form text that includes confidence statements about its own generation. Zhang & Choi (2023b) studied when uncertainty in predictions are caused by lack of knowledge or by lack of clarification.

Clarifying Question Prior works for training LLMs to ask clarifying questions have focused on settings where the types of clarification required are fixed by the task (i.e., there is fixed set of relevant input features (Wei et al., 2018; Kuhn et al., 2022)) or the input (e.g., reading-comprehension tasks where the context contains multiple correct answers (Guo et al., 2021)). Such works have studied clarifying question generation for a variety of tasks, including gauging social and moral situations (Pyatkin et al., 2022) and image classification (Yu et al., 2019).

Closely related to our work, Andukuri et al. (2024) trained LLMs to ask clarifying questions to resolve task ambiguity, where a single request may have multiple different indented tasks depending on the user, after user simulation. Compared to their experimental setting of simulated user persona for open-ended tasks, we address real-world open domain question answering task. Furthermore, we leverage preference optimization framework while they focus on generating supervised fine-tuning data by filtering low-quality clarifying questions. Hong et al. (2023) also uses LLM-simulated dialogues for training. Recent work has also noted the scarcity of clarifying questions in responses from existing LLMs, and have studied promoting methods for eliciting such responses with greater frequency (Deng et al., 2023a; Shaikh et al., 2023). Li et al. (2023) found that clarifying questions can be useful for learning individualized preferences.

7 Limitations and Future Work

We only consider systems for one or two turn interactions and does not consider cases where asking additional clarifying questions might be helpful after observing the user's clarifying answer. Future work might explore methods for extending our evaluation framework and double-turn preference annotation methods to accommodate general multi-turn interactions with users. Furthermore, the systems explored in this work also do not model dialogue acts (Stolcke et al., 2000) outside of predicting a single answer or asking a clarifying question. At times, it might be more appropriate for LLMs to generate responses for different behaviors (e.g., Overton responses containing multiple answers if there are only a few possible interpretations. Long vs. short answers for complex queries. Abstaining in lieu of predicting erroneous answers). Future work might consider how clarifying questions should be used in conjunction with other strategies. Another important direction is extending double-turn preference method for other tasks. Our evaluation framework is intended to be easily adaptable, only requiring (1) multiple annotator labels for expected outputs and (2) an changing end-task evaluation metric for the new setting.

8 Conclusion

We propose a method for training LLMs to ask clarifying questions with double-turn preferences and an an automatic framework for evaluating systems that ask clarifying questions using user simulated user interactions. Our QA results demonstrate that training systems to ask clarifying questions with double-turn preferences improves performance over training with standard single-turn preference labels and can be used to train models to judiciously decide when to ask for clarification.

Acknowledgments

We thank the UT NLP community for providing feedback throughout the project. This work was in part supported by Cisco Research. Any opinions, findings and conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of Cisco Research. The work is partially supported by a gift from Google Research. This work has taken place in part in the Rewarding Lab at UT Austin. The Rewarding Lab is supported by NSF (IIS-2402650), ONR (N00014-22-1-2204), EA Ventures, Bosch, UT Austin's Good Systems grand challenge, and Open Philanthropy. This work was done in part while the first and last author was visiting the Simons Institute for the Theory of Computing.

REFERENCES

Chinmaya Andukuri, Jan-Philipp Fränken, Tobias Gerstenberg, and Noah D Goodman. Star-gate: Teaching language models to ask clarifying questions. *ArXiv*, 2024. URL https://arxiv.org/pdf/2403.19154.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, T. J. Henighan, Nicholas Joseph, Saurav Kadavath, John Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom B. Brown, Jack Clark, Sam McCandlish, Christopher Olah, Benjamin Mann, and Jared Kaplan. Training a helpful and harmless assistant with reinforcement learning from human feedback. *ArXiv*, 2022. URL https://arxiv.org/abs/2204.05862.

Neil Band, Xuechen Li, Tengyu Ma, and Tatsunori Hashimoto. Linguistic calibration of language models. *ArXiv*, abs/2404.00474, 2024. URL https://api.semanticscholar.org/CorpusID:268819471.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. Reading Wikipedia to answer open-domain questions. 2017. URL https://arxiv.org/pdf/1704.00051.pdf.

Jeremy R Cole, Michael J.Q. Zhang, Daniel Gillick, Julian Martin Eisenschlos, Bhuwan Dhingra, and Jacob Eisenstein. Selectively answering ambiguous questions. *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023.

Yang Deng, Wenqiang Lei, Lizi Liao, and Tat seng Chua. Prompting and evaluating large language models for proactive dialogues: Clarification, target-guided, and non-collaboration. In *Findings of the Association for Computational Linguistics: EMNLP*, 2023a. URL https://arxiv.org/pdf/2305.13626.

Yang Deng, Wenxuan Zhang, Wai Lam, See-Kiong Ng, and Tat-Seng Chua. Plug-and-play policy planner for large language model powered dialogue agents. *ArXiv*, abs/2311.00262, 2023b. URL https://api.semanticscholar.org/CorpusID:264833170.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. Llm.int8(): 8-bit matrix multiplication for transformers at scale. *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2022. URL https://arxiv.org/pdf/2208.07339.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2024. URL https://arxiv.org/pdf/2305.14314.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *ArXiv*, 2024. URL https://arxiv.org/pdf/2407.21783.

Patrick Fernandes, Kayo Yin, Emmy Liu, André FT Martins, and Graham Neubig. When does translation require context? a data-driven, multilingual exploration. *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2021. URL https://arxiv.org/pdf/2109.07446.

Eve Fleisig, Rediet Abebe, and Dan Klein. When the majority is wrong: Modeling annotator disagreement for subjective tasks. In *Conference on Empirical Methods in Natural Language Processing*, 2023. URL https://api.semanticscholar.org/CorpusID:258615411.

Google Gemini Team. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. 2024. URL https://arxiv.org/pdf/2403.05530.

Meiqi Guo, Mingda Zhang, Siva Reddy, and Malihe Alikhani. Abg-coQA: Clarifying ambiguity in conversational question answering. In *3rd Conference on Automated Knowledge Base Construction*, 2021. URL https://openreview.net/forum?id=SlDZ1o8FsJU.

Joey Hong, Sergey Levine, and Anca Dragan. Zero-shot goal-directed dialogue via rl on imagined conversations. *arXiv preprint arXiv:2311.05584*, 2023.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022. URL https://arxiv.org/pdf/2106.09685.

Albert Jiang, Alexandre Sablayrolles, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, L élio Renard Lavaud, Louis Ternon, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mixtral of experts. *ArXiv*, 2024. URL https://arxiv.org/pdf/2401.04088.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. Language models (mostly) know what they know. *ArXiv*, 2022. URL https://arxiv.org/pdf/2207.05221.

Andreas Kopf, Yannic Kilcher, Dimitri von Rutte, Sotiris Anagnostidis, Zhi Rui Tam, Keith Stevens, Abdullah Barhoum, Nguyen Minh Duc, Oliver Stanley, Rich'ard Nagyfi, ES Shahul, Sameer Suri, David Glushkov, Arnav Dantuluri, Andrew Maguire, Christoph Schuhmann, Huu Nguyen, and Alexander Mattick. Openassistant conversations - democratizing large language model alignment. *ArXiv*, abs/2304.07327, 2023. URL https://api.semanticscholar.org/CorpusID:258179434.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Clam: Selective clarification for ambiguous questions with generative language models. *arXiv preprint arXiv:2212.07769*, 2022.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics (TACL)*, 2019. URL https://aclanthology.org/Q19-1026.pdf.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. Latent retrieval for weakly supervised open domain question answering. *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019. URL https://arxiv.org/pdf/1906.00300.

Yoonsang Lee, Xi Ye, and Eunsol Choi. Ambigdocs: Reasoning across documents on different entities under the same name. 2024. URL https://api.semanticscholar.org/CorpusID:269282807.

Belinda Z Li, Alex Tamkin, Noah Goodman, and Jacob Andreas. Eliciting human preferences with language models. *ArXiv*, 2023. URL https://arxiv.org/pdf/2310.11589.

Alisa Liu, Zhaofeng Wu, Julian Michael, Alane Suhr, Peter West, Alexander Koller, Swabha Swayamdipta, Noah A Smith, and Yejin Choi. We're afraid language models aren't modeling ambiguity. *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023. URL https://arxiv.org/pdf/2304.14399.pdf.

António V Lopes, M Amin Farajian, Rachel Bawden, Michael Zhang, and André FT Martins. Document-level neural mt: A systematic comparison. 2020.

Gemma Team Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, L. Sifre, Morgane Riviere, Mihir Kale, J Christopher Love, Pouya Dehghani Tafti, L'eonard Hussenot, Aakanksha Chowdhery, Adam Roberts, Aditya Barua, Alex Botev, Alex Castro-Ros, Ambrose Slone, Am'elie H'eliou, Andrea Tacchetti, Anna Bulanova, Antonia Paterson, Beth Tsai, Bobak Shahriari, Charline Le Lan, Christopher A. Choquette-Choo, Cl'ement Crepy, Daniel Cer, Daphne Ippolito, David Reid, Elena Buchatskaya, Eric Ni, Eric Noland, Geng Yan, George Tucker, George-Christian Muraru, Grigory Rozhdestvenskiy, Henryk Michalewski, Ian Tenney, Ivan Grishchenko, Jacob Austin, James Keeling, Jane Labanowski, Jean-Baptiste Lespiau, Jeff Stanway, Jenny Brennan, Jeremy Chen, Johan Ferret, Justin Chiu, Justin Mao-Jones,

Katherine Lee, Kathy Yu, Katie Millican, Lars Lowe Sjoesund, Lisa Lee, Lucas Dixon, Machel Reid, Maciej Mikula, Mateo Wirth, Michael Sharman, Nikolai Chinaev, Nithum Thain, Olivier Bachem, Oscar Chang, Oscar Wahltinez, Paige Bailey, Paul Michel, Petko Yotov, Pier Giuseppe Sessa, Rahma Chaabouni, Ramona Comanescu, Reena Jana, Rohan Anil, Ross McIlroy, Ruibo Liu, Ryan Mullins, Samuel L Smith, Sebastian Borgeaud, Sertan Girgin, Sholto Douglas, Shree Pandya, Siamak Shakeri, Soham De, Ted Klimenko, Tom Hennigan, Vladimir Feinberg, Wojciech Stokowiec, Yu hui Chen, Zafarali Ahmed, Zhitao Gong, Tris Brian Warkentin, Ludovic Peran, Minh Giang, Cl'ement Farabet, Oriol Vinyals, Jeffrey Dean, Koray Kavukcuoglu, Demis Hassabis, Zoubin Ghahramani, Douglas Eck, Joelle Barral, Fernando Pereira, Eli Collins, Armand Joulin, Noah Fiedel, Evan Senter, Alek Andreev, and Kathleen Kenealy. Gemma: Open models based on gemini research and technology. *ArXiv*, 2024. URL https://arxiv.org/pdf/2403.08295.

Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. AmbigQA: Answering ambiguous open-domain questions. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020. URL https://arxiv.org/pdf/2004.10645.pdf.

Yixin Nie, Xiang Zhou, and Mohit Bansal. What can we learn from collective human opinions on natural language inference data? *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020. URL https://arxiv.org/pdf/2010.03532.pdf.

OpenAI. Gpt-4 technical report. *ArXiv*, 2023. URL https://arxiv.org/pdf/2303.08774.pdf.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2022. URL https://arxiv.org/pdf/2203.02155.

Steven T Piantadosi, Harry Tily, and Edward Gibson. The communicative function of ambiguity in language. *Cognition*, 122(3):280–291, 2012.

Valentina Pyatkin, Jena D Hwang, Vivek Srikumar, Ximing Lu, Liwei Jiang, Yejin Choi, and Chandra Bhagavatula. Clarifydelphi: Reinforced clarification questions with defeasibility rewards for social and moral situations. *ArXiv*, 2022. URL https://arxiv.org/pdf/2212.10409.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2023. URL https://arxiv.org/abs/2305.18290.

Sebastin Santy, Jenny T Liang, Ronan Le Bras, Katharina Reinecke, and Maarten Sap. Nlpositionality: Characterizing design biases of datasets and models. In *Annual Meeting of the Association for Computational Linguistics*, 2023. URL https://api.semanticscholar.org/CorpusID:259076120.

Omar Shaikh, Kristina Gligorić, Ashna Khetan, Matthias Gerstgrasser, Diyi Yang, and Dan Jurafsky. Grounding or guesswork? large language models are presumptive grounders. *arXiv preprint arXiv:2311.09144*, 2023.

Andreas Stolcke, Klaus Ries, Noah Coccaro, Elizabeth Shriberg, Rebecca Bates, Daniel Jurafsky, Paul Taylor, Rachel Martin, Carol Van Ess-Dykema, and Marie Meteer. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational Linguistics*, 2000. URL https://aclanthology.org/J00-3003.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Elena Voita, Rico Sennrich, and Ivan Titov. When a good translation is wrong in context: Context-aware machine translation improves on deixis, ellipsis, and lexical cohesion. *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019. URL https://arxiv.org/pdf/1905.05979.pdf.

Zhilin Wang, Yi Dong, Olivier Delalleau, Jiaqi Zeng, Gerald Shen, Daniel Egert, Jimmy J. Zhang, Makesh Narsimhan Sreedhar, and Oleksii Kuchaiev. Helpsteer2: Open-source dataset for training top-performing reward models, 2024.

Wei Wei, Quoc Le, Andrew Dai, and Jia Li. Airdialogue: An environment for goal-oriented dialogue research. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 3844–3854, 2018.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi 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 M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP): System Demonstrations*, 2020. URL https://arxiv.org/pdf/1910.03771.

Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International conference on machine learning*, pp. 23965–23998. PMLR, 2022.

Lili Yu, Howard Chen, Sida Wang, Yoav Artzi, and Tao Lei. Interactive classification by asking informative questions. *ArXiv*, 2019. URL https://arxiv.org/pdf/1911.03598.

Yuewei Yuan, Chaitanya Malaviya, and Mark Yatskar. Ambicoref: Evaluating human and model sensitivity to ambiguous coreference. *Findings of the Association for Computational Linguistics: EACL*, 2023. URL https://arxiv.org/pdf/2302.00762.

Michael J.Q. Zhang and Eunsol Choi. SituatedQA: Incorporating extra-linguistic contexts into QA. *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2021. URL https://arxiv.org/pdf/2109.06157.pdf.

Michael J.Q. Zhang and Eunsol Choi. Clarify when necessary: Resolving ambiguity through interaction with lms. *ArXiv*, abs/2311.09469, 2023a. URL https://api.semanticscholar.org/CorpusID:265221465.

Michael J.Q. Zhang and Eunsol Choi. Clarify when necessary: Resolving ambiguity through interaction with lms. *ArXiv*, 2023b. URL https://arxiv.org/pdf/2311.09469.pdf.

Wenting Zhao, Xiang Ren, Jack Hessel, Claire Cardie, Yejin Choi, and Yuntian Deng. Wildchat: 1m chatgpt interaction logs in the wild. *International Conference on Learning Representations (ICLR)*, 2024. URL https://arxiv.org/pdf/2405.01470.

Kaitlyn Zhou, Jena D. Hwang, Xiang Ren, and Maarten Sap. Relying on the unreliable: The impact of language models' reluctance to express uncertainty. *ArXiv*, 2024a. URL https://arxiv.org/pdf/2401.06730.

Yifei Zhou, Andrea Zanette, Jiayi Pan, Sergey Levine, and Aviral Kumar. Archer: Training language model agents via hierarchical multi-turn rl, 2024b. URL https://arxiv.org/pdf/2402.19446.

Banghua Zhu, Evan Frick, Tianhao Wu, Hanlin Zhu, and Jiantao Jiao. Starling-7b: Improving llm helpfulness & harmlessness with rlaif, 2023. URL https://starling.cs.berkeley.edu/.

Question	NQ-Open Annotations Answers	AmbigQA Annotations Is Ambiguous?	Answers
Who is singing in the background of gold digger?	{"Ray Charles", "Jamie Foxx"}	No	{"Jamie Foxx"}
How much does a 2004 chevy cavalier weight?	{"1,162 - 1,315 kg", "2,562 - 2,900 lb" }	No	{"2,562–2,900 lb"}
What season does meredith and derek get married in grey's anatomy?	{"Season 7"}	Yes	{"Season 5", "Season 7"}
When did the soviet union test its first atomic bomb?	{"29 August 1949"}	Yes	{"29 August 1949", "7:00 a.m"}
When did the soviet union test its first atomic bomb?	{"29 August 1949"}	Yes	{"29 August 1949", "7:00 a.m"}

Table 5: QA examples differing labeled answer sets form NQ-Open and AmbigQA.

A Dataset Details

In Table 5, we provide examples from AmbigQA and NQ-Open with differing answer sets. These examples highlight instances where relying on a multiple annotators to provide the answer versus relying on a single annotator to provide all possible answers yielded differing answer sets.

A.1 Human and Model Ambiguity SFT and RLHF Training Dataset Details

In Table 7, we report dataset statistics for or two SFT datasets. Note that (x, q) pairs are used for SFT training for our clarifying question asking $(x \rightarrow q)$ systems. Additionally, the (x, q, a, y) pairs are used for training our joint and separate systems for predicting QA answers given clarification $(x, q, a \rightarrow y)$ as well as our user simulator models $(x, q, y \rightarrow a)$. We also use these pairs for training our direct QA systems $(x \rightarrow y)$, ignoring the clarifying question and answer. In Table 8, we include several examples from each dataset, highlighting the differences in the ambiguities identified in each.

For RLHF training, we use examples from the NQ-Open training set after removing examples used to generate our SFT datasets from NQ-Open. This leaves us with 70,904 remaining input questions, which we split into training and develpent splits (64,584 and 6,320).

B Experimental Details

We perform all experiments on a single machine with 8 A40 (48GB) GPUs using the transformers library (Wolf et al., 2020) and the AdamW optimizer (all training runs completed within 24 hours). During SFT training, we train all models with a learning rate of 5e-5 and batch size of 32. Training was performed for up to 5 epochs, evaluating on our development set after each epoch and selecting using the best performing checkpoint.

For DPO training, we merge LoRA checkpoints from our SFT-only baseline and train using a KL regularization factor of 0.1 and a learning rate of 5e-6 in all experiments. For training was performed for up to 2 epochs until loss converges on development data, selecting the best performing checkpoint on development data. For Llama2-7b based methods, we train with a batch size of 32, evaluating every 750 steps. For Gemma-7b based methods we train with a batch size of 16, evaluating every 1500 steps.

For performing additional SFT training to create our joint models for asking clarifying questions and answering with clarification, we merge LoRA checkpoints again and use the same hyperparameters used for standard SFT training listed above.

Input Query: Who is the highest paid football player?
GPT-4 Response (104 words): As of 2024, the highest-paid football players are Cristiano Ronaldo and Karim Benzema, both earning around €200 million per year. Ronaldo plays for Al Nassr, and Benzema is with Al Ittihad, both in the Saudi Pro League. Following them, N'Golo Kanté and Neymar each earn approximately €100 million annually, also playing in the Saudi Pro League for Al Ittihad and Al Hilal, respectively. Kylian Mbappé is the highest-paid player outside the Saudi Pro League, with a salary of €70 million at Paris Saint-Germain. These figures reflect the significant financial influence of the Saudi Pro League, which has attracted many top players with lucrative contracts.
Gemini Response (30 words): The highest-paid football player in the NFL for 2024 is Joe Burrow, the quarterback for the Cincinnati Bengals. He earns a whopping $55 million per year according to his contract.
Claude-3 Sonnet Response (112 words): As of 2023, the highest paid football (soccer) player in the world is Kylian Mbappé of Paris Saint-Germain. His reported annual salary is around €72 million (approximately $78 million). Some other top highest-paid football players in 2023 include: 2. Lionel Messi (Paris Saint-Germain) - Around €41 million per year 3. Neymar (Paris Saint-Germain) - Around €36.5 million per year 4. Kevin De Bruyne (Manchester City) - Around €25 million per year 5. Robert Lewandowski (Barcelona) - Around €23.5 million per year However, player salaries can fluctuate frequently with new contracts, bonuses and endorsements. Mbappé's new deal with PSG from 2022 made him the highest paid at the reported €72 million annual wages.

Table 6: LLM responses to the input query. All responses were generated on May 21st, 2024.

SFT Dataset	Train		Dev	
	(x, q)	(x, q, a, y)	(x, q)	(x, q, a, y)
Human-Identified Ambiguity	4000	8781	400	889
Model-Identified Ambiguity	4000	9189	400	948

Table 7: SFT Dataset Statistics.

Question (x)	Clarifying Question (q)	Clarifying Answers (a_i)	Answers (y_i)
where were the olympic games held in greece	Are you asking about the ancient Olympic Games or the modern Olympic Games?	Ancient Olympic Games / Modern Olympic Games	Olympia / Athens
who won the red army or the white army	Are you asking about the Russian Civil War or a specific battle?	Russian Civil War / Specific battle	The Red Army / the white army
who wrote ob la di ob la da	Are you asking about the primary composer or the co-writer?	Primary composer / Co-writer	Paul McCartney / John Lennon
original singer of how am i supposed to live without you	Are you asking about the original singer who first recorded the song or the singer who popularized it?	The original singer who first recorded the song / The singer who popularized the song	Laura Branigan / Michael Bolton

Table 8: Randomly sampled Model-Ambiguity SFT dataset examples.

Question (x)	Clarifying Question (q)	Clarifying Answers (a_i)	Answers (y_i)
Who played thanos in gardians of the galaxy?	Are you asking about the actor who provided the voice for Thanos, or the actor who physically portrayed him on screen?	The actor who provided the voice for Thanos. / The actor who physically portrayed him on screen.	Isaac C. Singleton Jr. / Sean Gunn
Where in the bible is the story of samuel?	Are you asking for the specific book and verses where the story of Samuel is found, or are you asking for the general section of the Bible where the story is located?	I'm asking for the specific book and verses. / I'm asking for the general section of the Bible.	1 Samuel 1:1–28 / Hebrew Bible
Most amount of points scored in an nba game?	Are you asking about the most points scored by a single player in a game or the highest combined score in a game?	Most points scored by a single player / Highest combined score in a game	100 / 370
Who has the highest instagram followers in the world?	Are you asking about the individual with the most followers on Instagram, or the account with the most followers overall?	The individual with the most followers / The account with the most followers overall	Cristiano Ronaldo / Instagram

Table 9: Randomly sampled Human-Ambiguity SFT dataset examples.

You will be given a question and several possible responses. Possible responses may be correct due to an ambiguity in the question or when the question was asked. Provide a clarifying question to determine which possible response is correct. If one or zero of the possible responses are correct, respond with "None". Provide your output in the following format: Clarifying Question: [Clarifying Question] 1. Clarifying Answer: [First Clarifying Answer] 1. Response: [Copied Response for First Clarifying Answer] 2. Clarifying Answer: [Second Clarifying Answer] 2. Response: [Copied Response for Second Clarifying Answer] ... Question: {question} Possible Responses: {answers}

Table 10: Prompt used for generating our Human and Model Ambiguity SFT datasets with GPT-4.

C Additional Baselines

Table 11 reports additional baselines on the Llama2 base model. The first three baselines adapt ideas from relevant recent work to our task setting, as described below.

C.1 Additional Baselines

ClarifyDPO (Ours w/ StarGaTE). This baseline follows the double-turn annotation scheme of ClarifyDPO (Ours) closely, differing only in how to assign preferences among clarifying questions. We adapt prior work, StarGaTE (Andukuri et al., 2024), which ranks clarifying questions to filter high-quality conversation traces to generate supervised fine-tuning data. We use their criterion, which is the likelihood of generating the gold answer after observing each simulated user's clarifying answer, averaged over all simulated users.

Proactive chain-of-thought (ProCoT) prompt Deng et al. (2023a) presents a chain-of-thought prompt method for instructing LLMs to first determine whether or not an input question is ambiguous before generating a direct-answer or clarifying question response. The need for instruction-following abilities for this baseline necessitates using the instruction-tuned variants of each base model. Therefore, performance of this approach may be inflated due to the stronger base model.

Ours w/ Plug-and-Play Dialogue Policy Planner (PPDPP) Deng et al. (2023b) proposes a SFT and RL methods for training a separate model predicting the proper dialogue act for the next assistant turn. Because their method is not designed to train systems to generate responses, only to decide which type or response to predict, we train a separate PPDPP model from our base LLM via SFT. This system is trained over the SFT training sets over the gold dialogue act: clarify if the question is ambiguous or directly answer otherwise. We do not include their RL method, as the authors report comparable performance with and without additional RL training. we found training to be hyper parameter sensitive, with the suggested hyperparameters training systems to always select a single option. Using this trained system, we then use its predictions to determine whether to predict a clarifying question or direct-answer response using our Clarify-or-Direct Ans DPO model.

Clarify-or-Direct Ans SFT We train a Clarify-or-Direct Ans SFT model by the union of two datasets used to train Direct Ans SFT and Clarify SFT above.

C.2 Discussion of Results

Table 11 reports main experimental results: the average turn involved (for efficiency) and answer F1 (for effectiveness). We find that training with preferences using answer likelihood (StarGaTE), which are also computed over double-turns, does not achieve the same improvements as using answer direct match (Ours). We hypothesize that this is due to the noisiness of comparing likelihoods

Table 11: Additional baselines (yellow) with the Llama2 base model. The main results are in Table 1.

Turn 2 Model	Avg Turns (↓)	Answer F1 (↑) Unambig	Ambig	Total
Direct-Ans SFT	1	0.254	0.168	0.211
Direct-Ans SFT (Sampled)	1	0.250	0.172	0.214
Clarify SFT	2	0.310	0.216	0.259
Clarify DPO (Starling)	2	0.310	0.257	0.283
Clarify DPO (Ours)	2	**0.383**	**0.282**	**0.328**
Clarify DPO (Ours w/ STaRGaTE)	2	0.302	0.239	0.272
Ours w/ ProCoT Prompt	1.65	0.197	0.129	0.156
Ours w/ PPDPP	1.54	0.277	0.214	0.239
Clarify-or-Direct-Ans SFT	1.12	0.256	0.184	0.213
Clarify-or-Direct-Ans DPO (Ours)	1.56	0.289	0.211	0.243

Table 12: Additional baselines (yellow) for Clarify-or-Answer methods. The main results are in Table 4. We report the percent of questions directly answered (DA%), and evaluate using F1, Direct-Answer (DA) Acc., and Ambig (A) Acc.

Method (DA%)	Answer F1	DA Acc	A Acc
Random (44%)	0.234	0.554	0.521
C-or-A Pred (44%)	**0.243**	**0.619**	0.537
Ours w/ ProCoT Prompt (0.35%)	0.156	0.601	0.492
Ours w/ PPDPP (0.46%)	0.239	0.577	**0.590**

to assigning preferences, which may result in preference labels between pairs of clarifying questions even when they are simply paraphrases of each other.

In the last block, we can compare models that sometimes ask clarifying question and sometimes directly answer the question. We also report an additional evaluation metric, Direct Answer Accuracy and Ambiguity Accuracy in Table 12. Clarify-or-Direct-Ans SFT underperforms Clarify-or-Direct-Ans DPO, showing that additioanl preference learning step yields meaningful gain. Despite using a stronger base model, the prompt-based method (ProCot Prompt) performs the worst, as it is not trained in-domain. Training a separate model for ambiguity prediction (Ours w/ PPDPP) performs well, showing almost 10% higher ambiguity accuracy than our full model. This is unsurprising as their approach optimizes for ambiguity prediction directly. However, our method still outperforms this model in answer accuracy. This demonstrates that our method (1) more frequently asks clarifying questions when it's useful to do so; and (2) avoids asking for clarification on ambiguous questions when the model is not able to successfully use the clarifying question. The relatively large gains on Direct-Answer accuracy compared to F1 indicate that we can expect performance on end-task metrics to further improve with better clarifying question generation.

D Example Outputs

We include examples of models successfully using clarifying questions to predict the target answers for ambiguous (in Table 15) and unambiguous test queries (in Table 14).

E Licensing

The transformers library is licensed under Apache License 2.0. AmbigQA does not list any license; however NaturalQuestions, the dataset which it is based on, is under Apache License 2.0. Llama2 is licensed under the LLAMA 2 Community License Agreement. Gemma is licensed under the Gemma Terms of Use. We license all our created assets under CC BY-NC-SA 4.0.

Pretend that you are a user asking an AI assistant a question. In response to your question, the AI assistant has asked you a clarifying question to help it determine which answer you expect. For each of the following expected answers, provide a clarifying answer to the AI assistant's clarifying question that indicates you are expecting that answer. All clarifying answers should be a concise sentence or phrase and should not contain the expected answer. If there is no possible clarifying answer, respond with "None." instead. Provid your response in the following format: Clarifying Answer 1: [Clarifying Answer for Expected Answer 1] Clarifying Answer 2: [Clarifying Answer for Expected Answer 2] ... Question: {question} Clarifying Question: {clarify_q} {answers}

Table 13: GPT-4 prompt used for simulating user clarifying answers during evaluation. All evaluations were carried out using the GPT-4 version gpt-4-0613.

Table 14: Example predictions where asking clarifying questions helped recover the correct answer to an unambiguous test query. Clarifying question predictions are taken from our LLama2-7B-based Clarify DPO (Ours) model trained with double-turn preferences. We also include the predicted LLama2-7B-based Direct-Ans SFT model predictions.

Question (x)	Pred Clarifying Question (q)	GPT-4 Clarifying Answers (a_i)	Clarify DPO Pred Ans. ($\widehat{y_i}$)	Direct-Ans SFT Pred Ans. ($\widehat{y_i}$)	Gold Ans (y_i)
When does telltale's guardians of the galaxy episode 3 come out?	Which platform are you asking about for Telltale's Guardians of the Galaxy Episode 3?	I'm asking about the release date on all platforms.	August 22 , 2017	October 17, 2017	August 22 , 2017
When did gods not dead 3 come out	Which "God's Not Dead 3" are you referring to? The first, second, or third installment?	I'm asking about the release date on all platforms.	March 30 , 2018	April 2, 2018	March 30 , 2018
When did the sea of monsters movie come out?	Which "Percy Jackson and the Sea of Monsters" movie are you asking about? The one released in 2013 or the one released in 2022?	The one released in 2013.	August 7 , 2013	March 26, 2010	August 7 , 2013
Who won the nobel prize for peace in 2014?	Which individual or organization won the Nobel Peace Prize in 2014?	The individuals who won	Kailash Satyarthi and Malala Yousafzai	Kailash Satyarthi	Kailash Satyarthi and Malala Yousafzai
How many islands are included in the federated states of micronesia?	Which specific islands are you referring to?	I'm referring to all the islands and islets that make up the country.	607 islands and islet	600	607 islands and islet

Table 15: QA-with-Clarification predictions on unambiguous test queries from our LLama2-7B model trained with double-turn preferences. We include examples where clarifying questions helped the model recover multiple annotated answers.

Question (x)	Pred Clarifying Question (q)	GPT-4 Clarifying Answers (a_i)	Pred & Gold Ans. ($\widehat{y_i}, \widehat{y_i}$)
How many starbucks ? are there around the world	As of which date are you asking about the number of Starbucks stores around the world?	As of the end of 2018. As of the end of 2004. As of the end of 2016. As of the end of 2017.	(28,218, 28,218) (6,000, 4,962) (23,768, 23,768) (25,000, 30,000)
Who sang the original do you love me?	Which version of "Do You Love Me" are you referring to? The original song by The Contours or the cover by The Partridge Family	The version that was a hit in the 1960s. The version from the musical Fiddler on the Roof.	(The Contours, The Contours) (Topol, Topol)
When did the queen became queen of england?	Which specific queen are you referring to?	Queen Victoria Queen Elizabeth II Queen Anne	(20 June 1837, 20 June 1837) (6 February 1952, 6 February 1952) (8 March 1702, 1 May 1707)