Clarify When Necessary: Resolving Ambiguity Through Interaction with LMs

Michael J.Q. Zhang and **Eunsol Choi**
Department of Computer Science
New York University
{michaelzhang,eunsol}@nyu.edu

Abstract

In this work, we explore the challenges of developing interactive assistants that resolve ambiguity by asking their users clarifying questions. Specifically, we develop a task-agnostic framework for evaluating a system's ability to determine when to ask for clarification. Determining when to ask for clarification is a challenging task that requires systems to consider the demands of the individual user (i.e., how much they prioritize speed and usability versus carefulness) and the distribution of interpretations for a given request (i.e., whether an ambiguous request has one dominant, inferable interpretation). Using this framework, we evaluate systems for determining when to clarify across three NLP applications: QA, MT, and NLI. Finally, we introduce present a novel uncertainty estimation approach, INTENT-SIM, that determines the utility of asking a clarifying question by estimating the entropy over user intents. Our method consistently outperforms existing uncertainty estimation approaches at identifying predictions that will benefit from clarification. Furthermore, we find that INTENT-SIM is robust, demonstrating improvements across a wide range of NLP tasks and LMs. Together, our work lays foundation for further studies on clarifying interactions with LM assistants.

1 Introduction

Ambiguity is embedded throughout natural language, and even simple utterances can have multiple interpretations when read in isolation. Ambiguity serves a key, communicative function in language, allowing speakers to omit details by relying on information that is inferable from the extra-linguistic context of the conversation (e.g., temporal, social, and physical) (Piantadosi et al., 2012). At times, however, the speaker's intent is still unclear despite the context. In such cases, further interaction is required to resolve the ambiguity, often by asking and answering clarifying questions.

With the recent progress in large language model (LLM) development, interactive AI assistants (e.g., ChatGPT, Claude, LLaMA-2) have risen to prominence in our daily lives; yet, these systems often fail to interact with users to resolve ambiguities in their requests. We identify that the inability to determine when to ask for clarification is one of core challenges limiting these systems' ability to ask clarifying questions. To address these challenges, we develop an framework for evaluating a system's ability to determine when to clarify and apply it to a variety of NLP tasks including question answering (QA), machine translation (MT), and natural language inference (NLI).

While most previous works on modeling ambiguity in NLP (particularly in QA and MT) have treated ambiguity as a binary classification task (i.e., does this input have multiple, valid interpretations?), determining when to ask for clarification requires systems to consider a variety of more nuanced factors, like whether an ambiguous utterances have one, dominant interpretation that can be inferred (e.g., "She's from Boston" typically does not mean "Boston, Georgia") and how much the individual user prioritizes the LLM's speed and usability versus its carefulness. Our evaluations capture these considerations by measuring a system's ability to maximize end-task performance while minimizing the interaction cost from the user.

To determine which examples will benefit from clarification, we evaluate this task as the first step in a three stage pipeline for resolving ambiguity, which we depict in Figure 1. Here, systems (1) determine if clarification is necessary before (2) asking a clarifying question and (3) responding given the user's clarifying answer. We also support these evaluations by deriving datasets of sampled interpretations of ambiguous inputs from existing datasets focused on modeling ambiguity in NLI, MT, and QA (Min et al., 2020; Bawden et al., 2018; Liu et al., 2023).

Findings of the Association for Computational Linguistics: NAACL 2025, pages 5541–5558
April 29 - May 4, 2025 ©2025 Association for Computational Linguistics

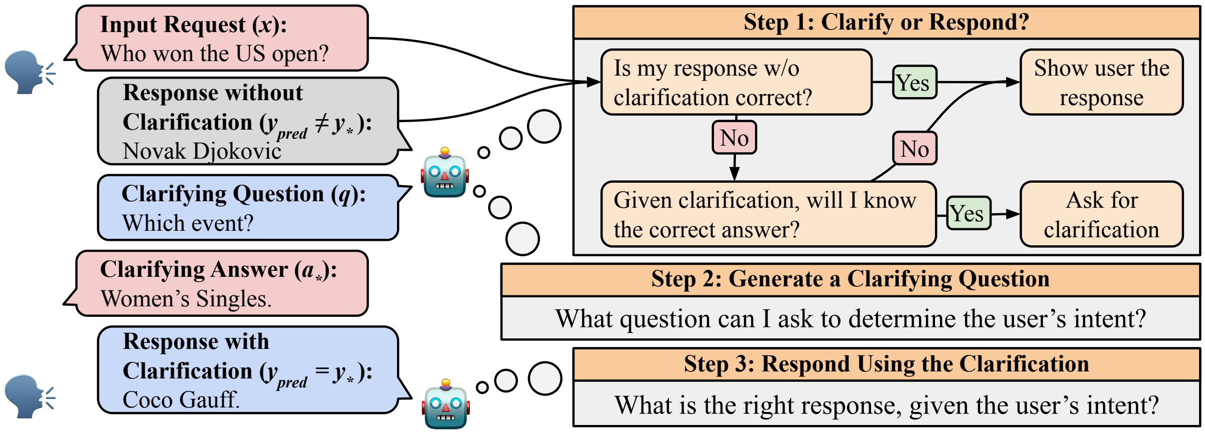
Input Request (x):
Who won the US open?
Response without Clarification ($y_{pred} \neq y_*$):
Novak Djokovic
Clarifying Question (q):
Which event?
Clarifying Answer (a_*):
Women's Singles.
Response with Clarification ($y_{pred} = y_*$):
Coco Gauff.
Step 1: Clarify or Respond?
Is my response w/o clarification correct?
Yes
Show user the response
No
No
Given clarification, will I know the correct answer?
Yes
Ask for clarification
Step 2: Generate a Clarifying Question
What question can I ask to determine the user's intent?
Step 3: Respond Using the Clarification
What is the right response, given the user's intent?

Figure 1: The three-stage pipeline for resolving ambiguity with clarification questions used in our evaluation framework. The first step is our target task, where systems must identify which inputs will benefit from clarification. In the second step, after deciding to clarify, we provide systems with a clarifying QA pair corresponding to the gold interpretation, which we generate from existing sources of disambiguated input/output pairs. Finally, in the third step, LLMs predict an output base on the input and the clarifying QA pair.

Finally, we conclude our work by introducing INTENT-SIM: a novel method for determining when to clarify. INTENT-SIM involves estimating the entropy over user intents by simulating multiple user-assistant interactions. Through our experiments, we demonstrate the INTENT-SIM consistently outperforms other uncertainty estimation baselines at identifying predictions that are incorrect and can be improved with clarification. We also find that these improvements are robust across different tasks and LLMs, with INTENT-SIM outperforms our other baselines at determining when to clarify in four out of six of the LLM-plus-task settings evaluated in this work.

2 Determining When to Clarify

We begin this work by introducing the task of determining when to ask for clarification. Later in this section, we dive into a formal description of our evaluation framework, which evaluates systems for determining when to clarify by using them as the first step in a three-step pipeline approach to resolving ambiguity with clarifying questions.

Setting Here, we provide some basic definitions for our setting and the three-step pipeline used for our evaluation framework depicted in Figure 1. We say that users provide an initial input request, x, to the LLM assistant. Some inputs may be ambiguous, resulting in many feasible output responses for the system to choose from, which we denote as $Y = \{y_i\}_1^k$. One of these outputs, $y_* \in Y$, represents the gold output corresponding to the user's intent behind their ambiguous request. To determine the users intent, systems may ask the user a clarifying question, q. The user then responds with the clarifying answer corresponding to their intent, $a_* \in A = \{a_i\}_1^k$. For simplicity, we assume a bipartite matching between the sets of clarifying answers, A, and feasible final responses, Y.

Each input, x, has its own distribution over intended interpretations given by $\mathbb{P}(y = y_*|x)$. Gathering annotations for the true distribution over intents is intractable and temperamental (i.e., subject to changes over time, location, and individual preferences); however, in Section 3 we describe several methods we use for generating our datasets of (x, y_*) tuples where y_* are realistic samples from the true distribution over intents.

2.1 Task Definition

Determining when to ask for clarification is a complex task that builds upon prior works on modeling ambiguity and estimating uncertainty in NLP. Existing approaches for modeling ambiguity in NLP applications has primarily focused on treating ambiguity as a binary label: either an input does or does not posses multiple valid interpretations (Min et al., 2020; Pilault et al., 2023). While it may seem natural to always ask for clarification for ambiguous inputs, the distribution over intended interpretations, given by $\mathbb{P}(y = y_*|x)$, is often dominated by a single most-likely interpretation. In such cases, it may be preferable to forego clarification and respond to the user's request directly.

How frequently systems should forego clarification for ambiguous requests depends on the demands of the domain and preferences of the user.

In high-stake settings, we may want systems to frequently ask for clarification. Likewise, for time-sensitive issues, we may want to minimize the number of interactions. As such, we define the task of determining when to clarify as an uncertainty estimation objective: where given an input request, x, systems must predict a scalar uncertainty estimate, $u(x)$, that correlates with how much performance is expected to improve after clarification.

This task differs from standard uncertainty estimation, where the objective is to produce uncertainty estimates that correlate with the correctness of a given prediction. In contrast, the objective of determining when to clarify is to identify predictions are both incorrect and will improve after clarification. This requires systems to disentangle the two factors that contribute to model uncertainty: epistemic and aleatoric uncertainty (Cole et al., 2023). Epistemic uncertainty refers to uncertainty that is due to a lack of knowledge. This may occur in QA with questions about an entities the LLM hasn't seen or in MT with words it hasn't observed the translation of. Aleatoric uncertainty, on the other hand, refers to uncertainty that is the result of some intrinsic randomness in the output. This randomness is often due to ambiguity, which we resolve through interaction. Systems for this task must identify instances with high aleatoric uncertainty, where the user's intent is ambiguous, and low epistemic uncertainty, where it has the knowledge required to respond after clarification.

2.2 Evaluation Framework

Following on the definition above, evaluating systems for determining when to clarify requires measuring the LLM assistant's end-task performance with and without access to clarification. To accomplish this, we model our evaluation framework after the three-stage pipeline depicted in Figure 1. We then evaluate systems by having them to perform the first step in this pipeline: determining which examples the LLM assistant should clarify and which it should respond to directly by predicting y_{pred}. Examples that the evaluated system decides to clarify are passed onto the later two stages the pipeline. In second step, we allow the LLM to ask a clarification question, q, to ask the user and receive their response, a_*. Finally, in the last step, the LLM assistant predicts y_{pred} based on the input and its clarification. The evaluation metrics defined below in Section 2.3 are then used to evaluate systems on their ability to maximize the performance of the LLM's predictions, y_{pred}, while minimizing the number of clarification queries made to the user.

In the setup above, the usefulness of asking for clarification is heavily dependent on (1) the quality of clarifying QA pair and (2) the LLM assistant's ability to utilize clarifications to determine the proper output. While several works have explored the task of generating good clarifying questions, particularly in classification (Yu et al., 2019), FAQ (Rao and Daumé III, 2018), and moral assessment (Pyatkin et al., 2022) domains, the task of determining when to ask for clarification has been comparatively under examined. As such, we opt to alleviate the dependence on the quality of the question generation system by using an oracle method for generating clarifying questions and answers. We describe this oracle system and the construction of the three-step ambiguity resolution pipeline in our evaluation framework in Section 4.

2.3 Evaluation Metrics

We define two metrics for evaluating uncertainty estimates, $u(x)$, for determining when to clarify:

Performance Under a Fixed Interaction Budget To evaluate uncertainty estimates for determining when to clarify, we provide systems with an interaction budget, $b \in [0, 100]$, and allow systems to ask clarification questions on $b\%$ of input examples. We use each system's uncertainty estimates, $u(x)$, to determine top $b\%$ of examples to provide clarification for, then evaluate system performance under this interaction budget. This metric is closely related to those used in selective prediction (El-Yaniv and Wiener, 2010), a uncertainty estimation task where low-confidence predictions are either withheld or passed onto a human oracle to annotate by hand (Tran et al., 2022).

AUROC This metric is commonly used in uncertainty quantification to evaluate an uncertainty estimator's ability to classify correct and incorrect predictions over all possible confidence thresholds. In our setting, we adapt this metric to evaluate the uncertainty estimate's ability to identify which predictions will improve from clarification.

3 Datasets and Applications

We apply our framework to three tasks and datasets for modeling ambiguity. All datasets label ambiguous inputs with their different interpretations, along with their respective outputs. We use these annotations later in developing our oracle system for

Task	Ambiguity Type	Input ($\boldsymbol{x}$) and Clarifying Question ($\boldsymbol{q}$)	Proportion
QA	Word-Sense Disambiguation / Entity Linking	$\boldsymbol{x}$: Who wins at the end of friday night lights? $\boldsymbol{q}$: Are you referring to the Friday Night Lights film, book, or television series?	48%
	Literal vs. Implied Interpretation	$\boldsymbol{x}$: Real name of gwen stacy in amazing spiderman? $\boldsymbol{q}$: Are you asking for the name of the actress who plays Gwen Stacy, or the full name of the character Gwen Stacy?	8%
	Multiple Valid Outputs	$\boldsymbol{x}$: When did west germany win the world cup? $\boldsymbol{q}$: Which time?	44%
NLI	Word-Sense Disambiguation	$\boldsymbol{x}$: Every night, the baby is fed milk. / Some nights, the baby is fed milk. $\boldsymbol{q}$: Does the baby get fed milk every night or just some nights?	44%
	Literal vs. Implied Interpretation	$\boldsymbol{x}$: The cake was so dry, it was like eating sand. / The cake was so dry, it was inedible. $\boldsymbol{q}$: Was the cake not suitable for eating or not safe to eat?	56%
MT	Word-Sense Disambiguation	$\boldsymbol{x}$: It's a little steeper than I was expecting. $\boldsymbol{q}$: What kind of mole are you referring to?	100%

Table 1: Example instances for each task for different ambiguity types, along with what proportion of ambiguities in each dataset fall into each type from our manual analysis on 150 examples.

generating clarifying questions and answers. All datasets lack existing labels for the distribution over these intents. Below, we describe each dataset in detail, as well as our methods for sampling intents for each example (further details in Appendix A).

3.1 Question Answering

We use the AmbigQA (Min et al., 2020) dataset, which re-annotates questions from NaturalQuestions (Kwiatkowski et al., 2019) with whether they are ambiguous. For each ambiguous example, they also annotate different intents as disambiguated revisions of the initial question paired their respective answers. We use the original annotated answers from NaturalQuestions as samples from the true distribution over intents, mapping these sampled outputs to their respective intents by identifying the disambiguation that contains the same answer.

QA Metric We evaluate QA using answer recall, measuring whether the gold answer string appears in the generated output after normalization (Chen et al., 2017). This deviates slightly from prior work (Rajpurkar et al., 2016) that evaluates strict exact match after normalization, as chat-based LLMs to generate verbose, sentence-length outputs as opposed to short answers (e.g., "The stern is the back of the boat." instead of "the back").

3.2 Natural Language Inference

We use the AmbiEnt dataset (Liu et al., 2023), which consists of ambiguous premise/hypothesis pairs that are paired with disambiguated revisions for each of their interpretations. Annotators for this dataset are first presented with the ambiguous input and are asked to label it as an NLI example. Annotators are then shown the different disambiguations each input, and are asked to label each interpretation again. We use these multiple annotations to identify which interpretation's label is consistent with the label annotators gave the initial, ambiguous input. We then use the matching interpretation as our sampled user intent and output label.

NLI Metric We evaluate NLI using 3-way (entails, contradicts, neutral) classification accuracy.

3.3 Machine Translation

Rich previous works have explored when sentence-level translation to fail in document-level context (Lopes et al., 2020; Yin et al., 2021; Voita et al., 2019). We source examples of ambiguous translations from one such work, DiscourseMT (Bawden et al., 2018), which manually curates a test set of ambiguous English-French translations. Each example consists of an ambiguous test sentence paired with two possible context sentences, where the translation of the test sentence depends on which context sentence precedes it. We use these test sentences, without context, as examples of ambiguous user inputs, taking its two translations as the set of feasible outputs. We also include the context sentences, which each have only one feasible translation, as examples of unambiguous inputs.

While this dataset does not contain annotations for estimating distribution over interpretations, sentences in this dataset are hand-crafted to be highly ambiguous. We, therefore, simply use the uniform distribution over interpretations in our experiments.

MT Metric We evaluate using contrastive accuracy (Maruf et al., 2019). This binary metric measures whether an LLM assigns a greater likelihood to the intended translation of an ambiguous sentence over the alternative. For unambiguous examples, we simply say that the system gets the interpretation correct without clarification. We deviate from the standard MT metrics (e.g., BLEU), as confounding factors such as variance in sentence structure often overshadow the word-level, semantic differences between translations.

3.4 Sources of Ambiguity Across Tasks

In Table 1, we provide analysis comparing the most common causes of ambiguity across tasks. The most common cause across all tasks is word-sense disambiguation. In QA, where named entities are more common, this commonly surfaces as entity linking ambiguities. The second cause is due to the literal and implied interpretations of each input. In QA, this usually occurs when a question literally means something different from what the user probably meant to ask. In NLI, we find this is often due to figurative language, where it is unclear whether the sentence should be interpreted literally. In MT, however, we find these ambiguities in the source sentence can usually be captured in its translation. The last common cause we find is ambiguity due to multiple valid outputs. This cause only affects QA reporting one of many answers may mislead users. We do not find this type of ambiguity in MT, where multiple translations of any sentence is a given, nor in NLI, where classes are mutually exclusive.

4 Ambiguity Resolution Pipeline

To evaluate systems for determining when to clarify, we first construct systems for performing the latter two steps of our three-step pipeline for resolving ambiguity through interaction.

Generating Oracle Clarifications To minimize the dependence on the LLM's ability to generate high quality clarifying questions, we use an oracle system for generating clarifying questions and answers. Our oracle makes use of few-shot prompting with GPT-3.5 (OpenAI, 2022), providing systems with instructions and two hand-written exemplars to accomplish the following task: Given the ambiguous input, x, and its different interpretations, each corresponding to a different output $y \in \{y_i\}_1^k$, systems must generate (1) clarifying question differentiating each interpretation, q, and (2) then clarifying responses, $\{a_i\}_1^k$, corresponding to each interpretation. Interpretations are provided in different formats depending on the available annotations in each dataset: we use disambiguated revisions of x for QA and NLI and the different target translations, $\{y_i\}_1^k$, for MT (details in Appendix B). Table 1 includes examples of generated clarification questions generated using this oracle.

Generating LLM Predictions To generate LLM predictions with and without providing clarification, we use standard four-shot prompting. We provide LLMs with demonstrations from the target task where exemplars include or do not include clarifications to match whether the test case does. We sample exemplars per-example and perform greedy decoding (exact prompts in Appendix E).

4.1 Do LLMs Utilize Clarifying Interactions?

Before evaluating systems for determining when to clarify, we must first establish that LLMs are capable of using clarifications and that providing clarifications can yield improvements on our derived datasets. To do this, we experiment with our ambiguity resolution pipeline by comparing the performance when opting to clarify no inputs versus clarifying all inputs. We also compare against not providing clarifying interactions, but instead providing the disambiguated versions of each input provided by each of the base datasets.

In addition to experimenting on our standard dataset of SAMPLED interpretations for ambiguous inputs, we also experiment with a UNIFORM version of our dataset which includes all the labeled interpretations of each input, where each is weighted each uniformly (note that for our MT dataset, these are equivalent due the construction process). While the standard SAMPLED setup is well suited for estimating system performance in realistic settings, it can also underestimate the importance of achieving high performance on uncommon intents. To avoid over-indexing on only the most common interpretations, we also evaluate on UNIFORM. Results here can help us determine whether LLMs are able to use clarifications for rare interpretations of inputs.

Model	Clarification	MT	QA		NLI	
		Uniform/Sampled	Uniform	Sampled	Uniform	Sampled
GPT3	Direct	50.0	22.7	51.8	31.2	41.7
	Clarify	85.8 (35.8)	40.8 (18.1)	61.8 (10.0)	31.6 (0.4)	45.9 (4.2)
	Disambig	84.7 (34.7)	41.2 (18.5)	62.0 (10.2)	30.6 (-0.6)	30.6 (-11.1)
LLAMA2 7B Chat	Direct	50.0	18.1	37.3	41.0	43.5
	Clarify	43.2 (-6.8)	32.0 (13.9)	47.9 (10.6)	55.3 (14.3)	52.5 (9.0)
	Disambig	44.9 (-5.1)	26.5 (8.4)	42.0 (4.7)	40.0 (-1.0)	40.0 (-3.5)
LLAMA2 13B Chat	Direct	50.0	17.9	40.0	28.0	40.7
	Clarify	40.9 (-9.1)	33.5 (15.6)	50.9 (10.9)	49.1 (21.1)	52.5 (11.8)
	Disambig	42.6 (-7.4)	28.5 (10.6)	45.2 (5.2)	26.6 (-1.4)	26.6 (-14.1)

Table 2: End-task results comparing three input settings: without clarification (Direct), with clarification (Clarify), and with the disambiguated input (Disambig). For QA and NLI, we evaluate under two different data generation processes, either uniformly weighing all interpretations or using our sampled interpretations. We evaluate MT using contrastive accuracy, QA using EM accuracy, and NLI using 3-way classification accuracy.

Results We report our results with using LLaMA-2-Chat and GPT-3 as the base LLM assistant in Table 2. We find that, across tasks and systems, LLMs can leverage clarifying questions and answers to improve their response. One exception to this trend, however, is the performance of LLaMA-2 variants on MT. We attribute this poor performance to LLaMA-2's low translation performance and insufficient multilingual pre-training (Touvron et al., 2023).

As expected, we also observe that clarification is often not necessary infer the correct interpretation and that models still frequently produce errors, even after clarification. These observations reinforce the challenges we highlighted above when determining when to clarify: systems must be able to identify when ambiguous inputs have a dominant, inferable interpretation, and they must be able to disentangle different forms of uncertainty to distinguish when incorrect predictions can and cannot be resolved through clarification.

Another notable trend is that systems tend to perform better with clarifying questions and answers than with disambiguated inputs, particularly for QA and NLI. We attribute this the way our QA and NLI datasets construct disambiguated interpretations. These datasets create disambiguated revisions of each ambiguous input by applying a minimal set of token-level edits to the initial input. While this makes disambiguations easier to annotate and compare, it comes at the cost of the naturalness of the resulting disambiguations. In contrast, our clarifying interactions do not have the same minimal-edit constraints and more closely resemble these systems' pretraining distributions.

5 Experiments

For our experiments on determining when to clarify, we use the same base LLMs as above for answering questions with and without clarification. We adapt existing methods for uncertainty estimation and chain-of-thought reasoning as baselines for this task. We begin this section by describing our novel approach, before introducing our baselines below.

5.1 INTENT-SIM

Unsupervised methods for uncertainty quantification in LLMs generally rely on estimating entropy over the output distribution, using high entropy to identify erroneous outputs (Kadavath et al., 2022; Kuhn et al., 2023). While these methods perform well at identifying incorrect predictions, they fail to identify *why* predictions are incorrect. Determining when to ask for clarification requires moving beyond simply predicting correctness and requires systems to identify when uncertainty is the result of ambiguity. In our proposed method, INTENT-SIM, we disentangle these two factors by explicitly estimating the ambiguity of a given input, which we quantify as the entropy over simulated user intents.

Table 3 illustrates our method. Using the same few-shot prompt structure for answering questions with clarification (exact prompt in Appendix E), we condition on the user's request to greedily generate a clarifying question. We then simulate different user intents by sampling multiple responses to the clarifying question (example generations in Table 3). Following Kuhn et al. (2023), we then cluster sets of semantically equivalent responses using a DeBERTa-large NLI model (He et al., 2021) finetuned on MNLI (Williams et al., 2018). We say that two responses are equivalent if either clarify-

Input with Sampled Clarification Question	Simulated User Answers	Likelihood
x_{MT}: There, on the trunk. q_{greedy}: What type of trunk are you referring to?	The large storage box at the back of a car. The large storage compartment of a car. The back of a car.	60%
	A large suitcase or box for storage. The large, wooden storage chest.	40%
x_{QA}: How many Grammy Awards does Whitney Houston have? q_{greedy}: Are you referring to the number of Grammy Awards Whitney Houston won, or the number of Grammy Awards Whitney Houston was nominated for?	The number of Grammy Awards Whitney Houston won. ***(Repeated* × *4)***	80%
	The number of Grammy Awards Whitney Houston was nominated for.	20%

Table 3: Generations from our Intent-Sim method. Systems greedily generate a clarifying question based on the input, then sample multiple user responses. We group equivalent responses using an NLI system, then compute the likelihoods and entropy over the grouped, simulated intents.

ing QA pair entails each other, then estimate the likelihood of each set as the proportion of samples in it. Finally, we compute our uncertainty estimate by computing the entropy of this distribution over semantically distinct answers. In our experiments we decode 10 user responses with $T = 0.5$ for all systems, following prior work on estimating uncertainty from samples (Cole et al., 2023) (details in Appendix C).

5.2 Baselines

Likelihood For this baseline, we prompt the model to generate the output without clarification using the same few-shot prompt as above. We then use the likelihood of the greedy output to determine when to clarify. This simple yet effective baseline is often used in uncertainty estimation for identifying incorrect model outputs. In this work, we use low-certainty in the output to identify where clarification may improve the model's response.

Self-Ask Introduced by Press et al. (2022), this prompting method elicits chain-of-thought reasoning from LLMs for compositional tasks such as multi-hop QA. In their method, LLMs decompose inputs into multiple sub-questions and answers, which are composed to get the final answer. Self-Ask uses an intermediate step, where models decide whether to continue generating sub-questions or generate the final response. We adapt this technique for our task, where the focus on querying for outside context, not on decomposing the input. We adjust our few-shot prompt from above to ask assistants after each input query "Is a follow-up question needed here?" (exact prompt in Appendix E). We then use the likelihood of generating "No" to score whether that clarification is needed. We also include this step in our few-shot exemplars, creating a 50-50 split between unambiguous inputs, where the system responds "No", and ambiguous inputs, where systems respond "Yes".

Sample Entropy Prior work in uncertainty estimation for LLMs has estimated the entropy over an LLM's output space by sampling multiple outputs and grouping equivalent responses (Kuhn et al., 2023; Cole et al., 2023). Following such works, we few-shot prompt the LLM to provide an output without clarification. We then group equivalent sampled responses using a pretrained NLI model (same as for Intent-Sim) to determine equivalent sampled QA outputs. For MT and NLI, we determine equivalence using exact match. We then cluster equivalent outputs and compute entropy over equivalent output sets (details in Appendix C).

5.3 Results

In Table 4, we report our results using different methods for deciding when to clarify. In comparing these systems against the random baseline, which randomly selects $b\%$ of examples to clarify and achieves percent gain in performance equal to b, we observe that Likelihood and Self-Ask demonstrate mixed results. While these systems generally perform better than random, they perform considerably worse than random under many settings. In contrast, using Sample Entropy and Intent-Sim consistently outperform all other baselines. In comparing just Sample Entropy and Intent-Sim, we find that Intent-Sim performs better in about two-thirds of the AUROC and budget settings we experiment with. Furthermore, note that in three out of the four settings where Intent-Sim achieved the best AUROC performance the dif-

Task	Model	Method	AUROC	$b = 10\%$	$b = 20\%$	$b = 30\%$
MT	GPT-3	Likelihood	**0.547**	76.1 (6%)	78.1 (17%)	79.8 (27%)
		Self-Ask	0.371	**77.3 (13%)**	**79.5 (25%)**	**81.5 (37%)**
		Sample Entropy	0.531	76.4 (11%)	78.4 (19%)	80.4 (30%)
		INTENT-SIM	0.512	**77.3 (13%)**	78.7 (21%)	79.3 (24%)
NLI	LLaMA-2 7B Chat	Likelihood	0.416	41.2 (1%)	40.0 (-7%)	39.4 (-11%)
		Self-Ask	0.477	41.6 (4%)	41.9 (7%)	42.5 (11%)
		Sample Entropy	0.467	43.9 (21%)	44.1 (22%)	**43.7 (19%)**
		INTENT-SIM	**0.531***	**44.3 (24%)**	**44.3 (24%)**	43.1 (15%)
	LLaMA-2 13B Chat	Likelihood	0.526	31.0 (14%)	33.0 (24%)	33.8 (27%)
		Self-Ask	0.462	28.2 (1%)	30.6 (12%)	34.0 (28%)
		Sample Entropy	0.525	29.8 (8%)	**33.0 (24%)**	33.8 (27%)
		INTENT-SIM	**0.544***	**31.0 (14%)**	32.8 (23%)	**34.8 (32%)**
QA	GPT-3	Likelihood	0.590	55.4 (14%)	55.9 (25%)	56.3 (35%)
		Self-Ask	0.538	55.1 (6%)	55.6 (18%)	56.2 (32%)
		Sample Entropy	0.625	**55.5 (17%)**	**56.1 (29%)**	**57.0 (49%)**
		INTENT-SIM	**0.628***	**55.5 (17%)**	**56.1 (29%)**	**57.0 (49%)**
	LLaMA-2 7B Chat	Likelihood	0.510	38.4 (-1%)	39.1 (14%)	39.7 (28%)
		Self-Ask	0.510	38.9 (10%)	39.3 (17%)	39.9 (32%)
		Sample Entropy	**0.532**	**39.1 (13%)**	**39.3 (19%)**	**40.1 (36%)**
		INTENT-SIM	0.501	38.7 (6%)	**39.3 (19%)**	39.7 (26%)
	LLaMA-2 13B Chat	Likelihood	0.551	41.1 (8%)	**41.7 (21%)**	41.8 (24%)
		Self-Ask	0.546	41.0 (6%)	41.6 (20%)	42.1 (30%)
		Sample Entropy	0.552	41.0 (6%)	41.4 (14%)	42.0 (28%)
		INTENT-SIM	**0.570**	**41.3 (11%)**	41.5 (17%)	**42.8 (37%)**

Table 4: Results for determining when to clarify. We report AUROC and performance under fixed interaction budget, b, evaluated using contrastive accuracy for MT, accuracy for QA and NLI. We also report the percent gain in performance relative to the total gain from clarifying all examples. (*) denotes cases where there was a statistically significant difference in AUROC with the best performing system when compared against all other baselines. We determine significance using $p < 0.05$ and 1,000 samples.

ference was statistically significant from all other baselines. This was not true for the two settings where other systems achieved the best AUROC.

6 Related Work

Clarifying Questions Fried et al. (2022) notes that a relationship between pragmatic reasoning and ambiguity, where ambiguity can often be resolved via pragmatic reasoning. Prior works explore such goal-oriented dialogues have studied task-specific settings. Shridhar et al. (2023) studies generating clarifying questions as a supervised learning task, and then using them for knowledge distillation. (Pyatkin et al., 2022) uses RL to guide their question generation models for moral judgments of a situations. Prior work (Yu et al., 2019) studies balancing asking clarification questions and making the final classification prediction over multi turn interactions. Their clarifying questions only cover existing attributes, while ours are open ended. Deng et al. (2023b) also notes the tendency for current LLMs to not ask clarifying questions, and explores methods to trigger such responses via prompting.

Clarifying questions have also been studied under the task of dialogue act prediction. While clarifying questions represent one class of dialogue acts, Deng et al. explores the task of predicting a variety of other dialogue acts that are relevant to other dialogue domains like negotiations. Deng et al. (2023a) surveys work exploring a wider range of 'dialogue strategies' and methods to predict when to use each. Our task of determining whether clarification is needed is closely related to such works on dialogue act prediction. Furthermore, Sorensen et al. (2024) describes the challenge and goal of developing pluralistically aligned language models that can accommodate a variety of user perspectives. Clarifying questions represent one such strategy, and our evaluation methods using sampled user interpretations are connected to the concept of distributional pluralism discussed in this work. Here, systems are evaluated on their ability to accommodate perspectives based on their distribution across the user population. We are able to develop an evaluation setting for this by relying on sampled interpretations to inputs where users may have diverse views and expect opposing outputs.

Uncertainty Estimation Several existing works have studied methods for disentangling different sources of uncertainty. Kamath et al. (2020) studies identifying out-of-distribution test examples, a source of epistemic uncertainty. Other works have studied uncertainty estimation techniques for LLMs (Kadavath et al., 2022; Lin et al., 2022), but they do not explicitly model or evaluate their ability to disentangle different sources of uncertainty. These works also explore supervised methods for uncertainty estimation in LLMs, but find that these methods generalize poorly to new domains.

Ambiguity in NLP Numerous datasets have been created for studying ambiguity in NLP, including work in coreference resolution (Yuan et al., 2023), NLI (Pavlick and Kwiatkowski, 2019), and MT (Pilault et al., 2023). The last work on MT also studies resolving ambiguity in an interactive chain-of-thought setting; however, it does not consider the challenge of modeling how ambiguous a given input is or determining whether interaction is helpful. (Parrish et al., 2021) has also used ambiguity resolution to study model biases, creating a task where systems are evaluated on if they resolve ambiguity by relying on harmful social biases.

7 Conclusion

We present a unified framework for resolving ambiguity with clarifying questions, applying it to QA, MT, and NLI. Our framework exposes the challenges in modeling clarifying interactions, and motivates the further study of disentangling uncertainty estimation and identifying when uncertainty can be attributed to ambiguity. We present a novel uncertainty estimation approach for this objective, INTENT-SIM, which we demonstrate improves detection of when to clarify.

8 Limitations

There is a computational overhead associated with our proposed INTENT-SIM uncertainty estimation method that comes from running an NLI model over samples from the LLM. In practice, we find that this additional cost is dominated by the need to sample multiple continuations from the LLM, and the overhead from running our NLI model or clustering algorithm are negligible in comparison.

While this work makes strides toward modeling ambiguity as a distribution over intents, rather than a binary classification as done in previous works on ambiguity in NLI and MT, we note that these distributions are highly dependent on a variety of extra-linguistic factors including time, location, and individual preferences. Further exploration is needed into exploring how this framework may be adapted to also account for such factors in both modeling and evaluation.

Finally, our work does not consider several other important challenges, including clarifying question generation, and handling arbitrary length interactions. We hope that our work will aid future efforts exploring these unaddressed challenges.

Acknowledgements

We thank the members of UT NLP community for early feedback on the draft, especially Jessy Junyi Li. 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.

References

Rachel Bawden, Rico Sennrich, Alexandra Birch, and Barry Haddow. 2018. Evaluating discourse phenomena in neural machine translation. In *16th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1304–1313. Association for Computational Linguistics (ACL).

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. In *55th Annual Meeting of the Association for Computational Linguistics, ACL 2017*, pages 1870–1879. Association for Computational Linguistics (ACL).

Jeremy R Cole, Michael JQ Zhang, Daniel Gillick, Julian Martin Eisenschlos, Bhuwan Dhingra, and Jacob Eisenstein. 2023. Selectively answering ambiguous questions. *arXiv preprint arXiv:2305.14613*.

Yang Deng, Wenqiang Lei, Wai Lam, and Tat-Seng Chua. 2023a. A survey on proactive dialogue systems: Problems, methods, and prospects. *arXiv preprint arXiv:2305.02750*.

Yang Deng, Wenqiang Lei, Lizi Liao, and Tat-Seng Chua. 2023b. Prompting and evaluating large language models for proactive dialogues: Clarification, target-guided, and non-collaboration. *arXiv preprint arXiv: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. In

The Twelfth International Conference on Learning Representations.

Ran El-Yaniv and Yair Wiener. 2010. On the foundations of noise-free selective classification. *J. Mach. Learn. Res.*, 11:1605–1641.

Daniel Fried, Nicholas Tomlin, Jennifer Hu, Roma Patel, and Aida Nematzadeh. 2022. Pragmatics in language grounding: Phenomena, tasks, and modeling approaches. *arXiv preprint arXiv:2211.08371*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

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

Amita Kamath, Robin Jia, and Percy Liang. 2020. Selective question answering under domain shift. *arXiv preprint arXiv:2006.09462*.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. *arXiv preprint arXiv:2302.09664*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Teaching models to express their uncertainty in words. *arXiv preprint arXiv:2205.14334*.

Alisa Liu, Zhaofeng Wu, Julian Michael, Alane Suhr, Peter West, Alexander Koller, Swabha Swayamdipta, Noah A Smith, and Yejin Choi. 2023. We're afraid language models aren't modeling ambiguity. *arXiv preprint arXiv:2304.14399*.

António V Lopes, M Amin Farajian, Rachel Bawden, Michael Zhang, and André FT Martins. 2020. Document-level neural mt: A systematic comparison. In *22nd Annual Conference of the European Association for Machine Translation*, pages 225–234.

Sameen Maruf, André FT Martins, and Gholamreza Haffari. 2019. Selective attention for context-aware neural machine translation. *arXiv preprint arXiv:1903.08788*.

Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2020. Ambigqa: Answering ambiguous open-domain questions. *arXiv preprint arXiv:2004.10645*.

OpenAI. 2022. Introducing chatgpt.

Alicia Parrish, Angelica Chen, Nikita Nangia, Vishakh Padmakumar, Jason Phang, Jana Thompson, Phu Mon Htut, and Samuel R Bowman. 2021. Bbq: A hand-built bias benchmark for question answering. *arXiv preprint arXiv:2110.08193*.

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

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

Jonathan Pilault, Xavier Garcia, Arthur Bražinskas, and Orhan Firat. 2023. Interactive-chain-prompting: Ambiguity resolution for crosslingual conditional generation with interaction. *arXiv preprint arXiv:2301.10309*.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A Smith, and Mike Lewis. 2022. Measuring and narrowing the compositionality gap in language models. *arXiv preprint arXiv:2210.03350*.

Valentina Pyatkin, Jena D Hwang, Vivek Srikumar, Ximing Lu, Liwei Jiang, Yejin Choi, and Chandra Bhagavatula. 2022. Reinforced clarification question generation with defeasibility rewards for disambiguating social and moral situations. *arXiv preprint arXiv:2212.10409*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Conference on Empirical Methods in Natural Language Processing*.

Sudha Rao and Hal Daumé III. 2018. Learning to ask good questions: Ranking clarification questions using neural expected value of perfect information. *arXiv preprint arXiv:1805.04655*.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2021. Learning to retrieve prompts for in-context learning. *arXiv preprint arXiv:2112.08633*.

Kumar Shridhar, Alessandro Stolfo, and Mrinmaya Sachan. 2023. Distilling reasoning capabilities into smaller language models. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 7059–7073, Toronto, Canada. Association for Computational Linguistics.

Taylor Sorensen, Jared Moore, Jillian Fisher, Mitchell Gordon, Niloofar Mireshghallah, Christopher Michael Rytting, Andre Ye, Liwei Jiang, Ximing Lu, Nouha Dziri, et al. 2024. A roadmap to pluralistic alignment. *arXiv preprint arXiv:2402.05070*.

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

Dustin Tran, Jeremiah Liu, Michael W Dusenberry, Du Phan, Mark Collier, Jie Ren, Kehang Han, Zi Wang, Zelda Mariet, Huiyi Hu, et al. 2022. Plex: Towards reliability using pretrained large model extensions. *arXiv preprint arXiv:2207.07411*.

Elena Voita, Rico Sennrich, and Ivan Titov. 2019. When a good translation is wrong in context: Context-aware machine translation improves on deixis, ellipsis, and lexical cohesion. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1198–1212.

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

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

Kayo Yin, Patrick Fernandes, André FT Martins, and Graham Neubig. 2021. When does translation require context? a data-driven, multilingual exploration. *arXiv preprint arXiv:2109.07446*.

L. Yu, Howard Chen, Sida Wang, Yoav Artzi, and Tao Lei. 2019. Interactive classification by asking informative questions. *ArXiv*, abs/1911.03598.

Yuewei Yuan, Chaitanya Malaviya, and Mark Yatskar. 2023. Ambicoref: Evaluating human and model sensitivity to ambiguous coreference. *arXiv preprint arXiv:2302.00762*.

A Data Details

In Table 8 include of raw examples from each of our MT, QA, and NLI datasets and in Table 6 we include dataset statistics, including the number of ambiguous versus unambiguous examples, and the total number of interpretations. In matching interpretations from NaturalQuestions to AmbigQA disambiguations, we eliminate all examples where the NaturalQuestions answers do not appear in any of the AmbigQA interpretations and when it matches more than one interpretation.

B Oracle Question Generation System Details

We present our oracle clarification generation prompts for QA, NLI, and MT in Table 9, Table 10, and Table 11, respectively. We do not provide GPT-3.5 any system prompt, and the entire body of the prompt is provided in a single user-side message. Note that for our MT oracle prompt, there is the risk of answer leakage, since the output translation is included in the prompt. However, we do not find this is an issue, as the generated followup questions and answers are always in the source language only.

C Modeling Details

In Figure 2, we outline the INTENT-SIM algorithm in detail. Note that we greedily sample the clarifying question before sampling user responses at temperature $T = 0.5$. Note that, as with all sample-based approaches used in this work, we do not edit the exemplars between different generation steps or samples. Note that for our Sample Entropy baselines, we use a similar DFS strategy as depicted in Figure 2 lines 11-16 for clustering equivalent outputs and estimating entropy over the sampled outputs. We also use the same sampling hyperparameters as for INTENT-SIM. Inference was done on 4 NVIDIA A40 GPUS and the largest LLAMA-2 13B experiments took less than 5 hours each. We use the Transformers (Wolf et al., 2020) library for our implementations of LLAMA-2 models and variatns

In our initial testing, we find that model predictions can be sensitive to the choice in few-shot examples. In preliminary experiments, we explored sampling greedy answers from the LLMs to 20000 questions from NaturalQuestions over 10 randomly selected sets of few-shot exemplars using Llama2. We then counted the number of unique answers predicted for each question (after normalizing for whitespace, casing, etc.). In Table 5, we provide statistics of the number of unique answers generated. Given this variation, we evaluate all methods by randomly sample a new set of exemplars for each test example to ensure that our evaluations are not sensitive to any particular choice in few-shot exemplars. Prior work such as Rubin et al. (2021) has also explored the topic of varying few-shot exemplars for in-context learning. We will further clarify this point in our revisions.

# of Unique Answers	% of Questions
1	29%
2	24%
3	17%
4	11%
>5	18%

Table 5: Statistics of the number of unique answers generated per questions from sampling greedy answers from the LLMs to 20000 questions from NaturalQuestions over 10 randomly selected sets of few-shot exemplars using Llama2.

```
Input: LM $M$, NLI model $N$, User input $\mathbf{x}$, sampling temperature $T$, and simulation count $S$.
Output: Entropy over simulated intents, $u$.
 1: $\mathbf{q} \leftarrow$ GreedySample$(M, [\mathbf{x}])$
 2: for $i \in \{1, \ldots, S\}$ do
 3:     $\mathbf{a_i} \leftarrow$ TempSample$(M, [\mathbf{x}; \mathbf{q}], T)$
 4: $G \leftarrow \emptyset$
 5: for $i \in \{1, \ldots, S-1\}$ do
 6:     for $j \in \{i+1, \ldots, S\}$ do
 7:         left $\leftarrow N([\mathbf{q}; \mathbf{a_i}], [\mathbf{q}; \mathbf{a_j}])$
 8:         right $\leftarrow N([\mathbf{q}; \mathbf{a_j}], [\mathbf{q}; \mathbf{a_i}])$
 9:         if left is entailment or right is
            entailment then
10:             $G \leftarrow G \cup \{<i, j>, <j, i>\}$
11: $C \leftarrow \emptyset$
12: for $i \in \{1, \ldots, S\}$ do
13:     if $\mathbf{a_i} \notin c \quad \forall c \in C$ then
14:         $C \leftarrow C \cup$ DFS$(G, a_i)$
15: $\widehat{P}(c|\mathbf{x}) \leftarrow \frac{|c|}{S}, \quad \forall c \in C$
16: $u \leftarrow$ Entropy$(\widehat{P}(\cdot|\mathbf{x}))$
```

Figure 2: INTENT-SIM algorithm. We sample a clarifying question and responses from the LLM. We then construct a equivalence graph of responses, G, using NLI to determine equivalence. Finally, we identify disjoint subgraphs of G with depth-first-search, representing distinct intents, and estimate the entropy over intents.

D Additional Responsiveness to Clarification Results

In Table 7, we report our full results on responsiveness to clarifications. In addition to what we report in the main paper in Table 2, we also include results on LLAMA2 variants without chat-finetuning. Most notably from these additional results, we can see that chat-finetuning has strong effect on the overall performance across all tasks; however, the absolute gains from providing clarification only to increase in our NLI evaluations, and not notably in our QA results. This suggests that these clarification dialogue strategies may not be suffieicntly learned in these current chat-finetuning approaches.

E Prompts

We present the prompts for responding with clarification, without clarification, and for SelfAsk in Tables 12, 14, and 13. These tables also demonstrate the variations in prompt between tasks, particularly in the instructions. We base our NLI instruction and class-to-token mapping on the prompts from (Liu et al., 2023).

To perform our experiments with disambiguated inputs for QA and NLI, we use the same prompt as responding without clarification, substituting the input with the disambiguate form of the input. For MT where disambiguations are given as additional context sentences, we simply prepend "Context: ..." onto each user input, filling in the context sentence.

For sampling unambiguous examples for Self-Ask, we use the unambiguous examples labeled in the MT and QA datasets. For NLI, where all examples are labeled as ambiguous, we use examples where all 9 annotators interpreted the input the same way as unambiguous examples, as these demonstrate the least variation in user intents.

F Licensing

AmbigQA does not list any license; however NaturalQuestions, the dataset which it is based on, is under the Apache License 2.0. The DiscourseMT dataset is licensed under CC BY-SA 4.0. LLAMA2 is licensed under the LLAMA 2 Community License Agreement.

G Ethical Considerations

We do not collect any data in this paper. While we do generate datasets, we visually inspect generated examples and do not find instances of harmful or offensive content. The datasets we use in this work have been previously vetted by their authors in prior work. We also note that our work is only applied to English QA and NLI datasets and English-French translation.

Task	Ambiguous x	Unambiguous x	Sampled Interpretations y_*	Total Interpretations y_*
NLI	504	0	504	1008
QA	652	830	1482	2781
MT	88	176	352	352

Table 6: Counts of ambiguous and unambiguous inputs for each task. We also include counts of the number of sampled interpretations used in our "determining when to clarify" evaluations and the total number of interpretations.

Model	Clarification	MT	QA		NLI	
		Uniform/Sampled	Uniform	Sampled	Uniform	Sampled
GPT3	Direct	50.0	22.7	51.8	31.2	41.7
	Clarify	85.8 (35.8)	40.8 (18.1)	61.8 (10.0)	31.6 (0.4)	45.9 (4.2)
	Disambig	84.7 (34.7)	41.2 (18.5)	62.0 (10.2)	30.6 (-0.6)	30.6 (-11.1)
LLAMA2 7B	Direct	50.0	14.5	31.4	29.4	32.4
	Clarify	46.6 (-3.4)	27.3 (12.8)	45.4 (14.0)	25.4 (-4)	35.9 (3.5)
	Disambig	45.5 (-4.5)	25.7 (11.2)	41.1 (9.7)	29.8 (0.4)	29.8 (-2.6)
LLAMA2 7B Chat	Direct	50.0	18.1	37.3	41.0	43.5
	Clarify	43.2 (-6.8)	32.0 (13.9)	47.9 (10.6)	55.3 (14.3)	52.5 (9.0)
	Disambig	44.9 (-5.1)	26.5 (8.4)	42.0 (4.7)	40.0 (-1.0)	40.0 (-3.5)
LLAMA2 13B	Direct	50.0	17.7	39.1	30.6	37.4
	Clarify	46.6 (-3.4)	34.1 (16.4)	53.7 (14.6)	34.6 (4.0)	43.1 (5.7)
	Disambig	47.2 (-2.8)	32.4 (14.7)	50.8 (11.7)	30.2 (-0.4)	30.2 (-7.2)
LLAMA2 13B Chat	Direct	50.0	17.9	40.0	28.0	40.7
	Clarify	40.9 (-9.1)	33.5 (15.6)	50.9 (10.9)	49.1 (21.1)	52.5 (11.8)
	Disambig	42.6 (-7.4)	28.5 (10.6)	45.2 (5.2)	26.6 (-1.4)	26.6 (-14.1)

Table 7: Full Responsiveness to clarification results. Here, we alos inlude results of the base LLAMA-2 systems without chat finetuning. We evaluate three input settings: with clarification (Clarify), with the disambiguated input (Disambig), and baseline (Direct) without clarification. For QA and NLI, we evaluate under two different data generation processes, either uniformly weighing all interpretations or using our sampled interpretations. We evaluate MT using contrastive accuracy, QA using EM accuracy, and NLI using 3-way classification accuracy.

Task	Input (x)	Interpretations / Outputs (y)
MT	That is so sweet!	**Context:** You've been so wonderful to me these past couple of months. **Target:** C'est tellement adorable.
		Context: Try some - it's like a sugar explosion! **Target:** C'est tellement sucré.
	I've never seen so much dough!	**Context:** The pizza's in the oven, but there's still some dough left. **Target:** Je n'ai jamais vu autant de pâte !
		Context: Here you are - you've earnt it. **Target:** Je n'ai jamais vu autant de thune !
QA	When is episode 113 of dragon ball super coming out?	**Disambig:** When is episode 113 of dragon ball super coming out for its original airdate? **Answer:** October 29, 2017
		Disambig: When is episode 113 of dragon ball super coming out for its american airdate? **Answer:** June 1, 2019
	Who plays the science officer on star trek discovery?	**Disambig:** Who plays the science officer on star trek discovery who is a chief engineer? **Answer:** Anthony Rapp
		Disambig: Who plays the science officer on star trek discovery who is a Kelpien? **Answer:** Doug Jones
		Disambig: Who plays science officer Michael Burnham on Star Trek Discovery? **Answer:** Sonequa Martin-Green
NLI	A large number of people were not willing to take the risk. / A small number of people were willing to take the risk.	**Disambig:** A large number of people, but not all people, were not willing to take the risk. **Label:** entailment
		Disambig: A large number of people, and possibly all people, were not willing to take the risk. **Label:** neutral
	We have not been able to find any scientific evidence that extraterrestrial life exists. / There is no scientific evidence that extraterrestrial life exists.	**Disambig:** There is no scientific evidence to be found that extraterrestrial life exists. **Label:** neutral
		Disambig: There has been no scientific evidence collected that extraterrestrial life exists. **Label:** entailment

Table 8: Raw ambiguous examples from each dataset.

Given the Ambiguous Question and several possible Intended Interpretations, ask a Clarification Question and provide Clarification Responses corresponding to each Intended Interpretations. Here are two examples:

Example 1:
Ambiguous Question: Who has the highest goals in world football?
Intended Interpretation 1: Who has the highest goals in men's world international football?
Intended Interpretation 2: Who has the highest goals all-time in men's football?
Intended Interpretation 3: Who has the highest goals in women's world international football?

Clarification Question: Are you referring to the highest goals in men's world international football, or the highest goals in women's world international football?
Clarification Response 1: The highest goals in men's world international football.
Clarification Response 2: The highest goals all-time in men's football.
Clarification Response 3: The highest goals in women's world international football.

Example 2:
Ambiguous Question: Who won the last olympic men's hockey?
Intended Interpretation 1: Who won Olympic men's ice hockey in 2014?
Intended Interpretation 2: Who won Olympic men's ice hockey in 2010?
Intended Interpretation 3: Who won Olympic men's ice hockey in 2006?
Intended Interpretation 4: Who won the 2016 olympic men's field hockey?
Intended Interpretation 5: Who won the 2012 olympic men's field hockey?
Intended Interpretation 6: Who won the 2008 olympic men's field hockey?
Clarification Question: Which year? Are referring to field hockey or ice hockey?
Clarification Response 1: 2014, ice hockey.
Clarification Response 2: 2010, ice hockey.
Clarification Response 3: 2006, ice hockey.
Clarification Response 4: 2016, field hockey.
Clarification Response 5: 2012, field hockey.
Clarification Response 6: 2008, field hockey.

Now do it yourself:
Ambiguous Question: {}
Intended Interpretation 1: {}
...
Intended Interpretation k: {}

Table 9: QA Followup Generation Prompt.

Given the Ambiguous Phrase and two possible Intended Interpretations, ask a Clarification Question and provide two Clarification Responses corresponding to each Intended Interpretations. Here are two examples:

Example 1:
Ambiguous Phrase: Jon will wash his car, and Mary will too.
Intended Interpretation 1: Jon will wash his car, and Mary will wash hers.
Intended Interpretation 2: Jon and Mary will both wash Jon's car.
Clarification Question: Will Jon and Mary wash the same or different cars?
Clarification Response 1: The same.
Clarification Response 2: Different.

Example 2:
Ambiguous Phrase: The hospital is being sued by six foot doctors.
Intended Interpretation 1: The hospital is being sued by six podiatrists.
Intended Interpretation 2: The hospital is being sued by doctors who are six feet tall.
Clarification Question: Do you mean six podiatrists or doctors who are six feet tall.
Clarification Response 1: Podiatrists.
Clarification Response 2: Doctors who are six feet tall.

Now do it yourself:
Ambiguous Phrase: {}
Intended Interpretation 1: {}
Intended Interpretation 2: {}

Table 10: NLI Followup Generation Prompt.

```
Given the Ambiguous Phrase and two possible Translations, ask a Clarification Question about the meaning of a specific
word or phrase and provide two Clarification Responses corresponding to each Translation. Here are two examples:

Example 1:
Ambiguous Phrase: Where are the bats?
Translation 1: Où sont les chauves-souris?
Translation 2: Où sont les battes?
Clarification Question: What type of bats do you mean?
Clarification Response 1: The small animals with wings.
Clarification Response 2: Sticks like you would use for a sport.

Example 2:
Ambiguous Phrase: I love dates.
Translation 1: J'adore les rendez-vous.
Translation 2: J'adore les dattes.
Clarification Question: What type of dates are you referring to?
Clarification Response 1: Meetings or appointments with someone.
Clarification Response 2: The sweet fruit from the date palm tree.

Now do it yourself:
Ambiguous Phrase: {}
Translation 1: {}
Translation 2: {}
```

Table 11: MT Followup Generation Prompt.

Role	Messages
System	Translate the English sentences into French.
User	English: It's tight enough if you ask me.
Assistant	Follow-Up Question: Are you referring to physical tightness or tension in a situation?
User	Follow-Up Response: I'm referring to a tense or stressful situation.
Assistant	French: C'est déjà assez tendu à mon avis.
...	
User	English: Why don't you care for them then?
Assistant	Follow-Up Question: What do you mean by 'care for them'?
User	Follow-Up Response: I mean, why don't you take responsibility for them or look after them?
Assistant	French: Alors pourquoi tu ne t'en occupes pas ?

Table 12: Prompt for few-shot MT with clarification questions. The colored text represents the output generated from the LLM assistant.

Role	Messages
System	Answer the question.
User	Question: How many medals did australia win in the 2000 olympics?
Assistant	Is a Follow-Up Question Needed Here? Yes.
	Follow-Up Question: Are you asking about the number of gold, silver, bronze, or total medals that Australia won in the 2000 Olympics?
User	Follow-Up Answer: The total number of medals that Australia won in the 2000 Olympics.
Assistant	Answer: 58.
User	Question: Who plays erica on the last man on earth?
Assistant	Is a Follow-Up Question Needed Here? No.
	Answer: Cleopatra Coleman.
...	
User	Question: How long do contestants get to answer on jeopardy?
Assistant	Is a Follow-Up Question Needed Here? No.

Table 13: Prompt for few-shot QA with SelfAsk. The colored text represents the output that is scored by LLM assistant, and used as an uncertainty estimate. Sampled exemplars are always half unambiguous examples, not requiring follow-up questions, and half ambiguous examples, requiring follow-ups.

Role	Messages
System	For each Context, determine whether the Claim is True, False, or Inconclusive.
User	Context: The professor admitted that no students wrote course evaluations, which is surprising. Claim: It is surprising that no students wrote course evaluations.
Assistant	Answer: True.
...	
User	Context: Many birds are attracted to the island because of the abundance of food. Claim: Many birds are attracted to the island because of the abundance of trees.
Assistant	Answer: Inconclusive.

Table 14: Few-shot NLI prompt without clarification. The colored text represents the output generated from the LLM assistant.