
What Is a Clarification? Estimating Model-Dependent Input Ambiguity

Jingqin Tang Tim Tomov Sebastian Schmidt
Dominik Fuchsgruber Stephan Günnemann

TUM School of Computation, Information and Technology
Technical University of Munich, Germany

{go39gov,tim.tomov,sebastian95.schmidt,d.fuchsgruber,s.guennemann}@tum.de

Abstract

Ambiguous inputs can admit multiple interpretations, each potentially yielding a different answer. Existing work largely treats this ambiguity as a property of the input, yet for a particular language model, only some alternatives may be known and answerable. We therefore distinguish world ambiguity, capturing all alternatives admitted by an input, from model ambiguity, capturing those accessible to the responding model. To characterize these alternatives, we formalize natural-language clarifications through three desiderata: *uniqueness*, *faithfulness*, and *answer equivalence*. We show that these desiderata induce a one-to-one correspondence between valid answers and equivalence classes of clarifications, implying that any interpretation can be represented either by its answer or by the clarifications that uniquely express it. We leverage this correspondence to introduce CIRCLECLAR, a black-box method that estimates the set of answers available to a model by checking consistency along an answer–clarification–answer cycle. Across open-domain QA and executable Text-to-SQL, CIRCLECLAR achieves strong recovery of these answer sets. In QA, the recovered support yields the strongest overall performance in model-dependent ambiguity detection and enables effective selection among answering, clarifying, and abstaining. Together, these results demonstrate the practical value of a model-dependent view of ambiguity and its characterization through an answer-clarification correspondence.

1 Introduction

Ambiguous inputs can admit multiple reasonable interpretations, each potentially leading to a different answer. Existing work has largely treated this ambiguity as a property of the input itself [12, 14, 4, 5, 11, 6, 16]. However, when an ambiguous input is presented to a Large Language Model, not every valid interpretation is necessarily accessible and answerable by that model. Consider, for example, the question “*Who voiced the snake in the Jungle Book?*”. The intended adaptation is unspecified: for the 1967 film, the answer is Sterling Holloway, whereas for the 2016 film, it is Scarlett Johansson. While both interpretations are valid, one model may know both films and their answers, whereas another may know only one. The ambiguity inherent in the question remains unchanged; what changes is *how much of that ambiguity is accessible to the responding model*.

We therefore propose to distinguish between *world ambiguity* and *model ambiguity* (Figure 1). World ambiguity captures the alternatives admitted by the input, independent of any particular model. Model ambiguity instead captures those alternatives that are *known and answerable* to the responding model. The distinction is consequential: two models can face the same world ambiguity but, because they know different parts of its ambiguity, effectively face different problems. A model that knows several valid alternatives should recognize that committing to one leaves the user’s intent unresolved; a

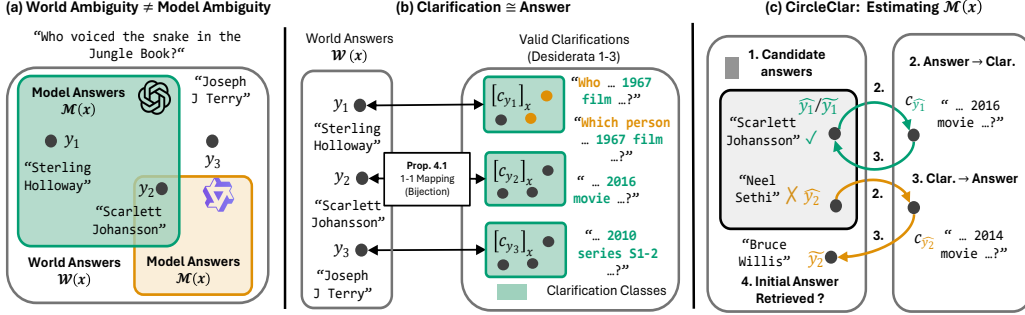


Figure 1: **Overview of our framework:** illustrated with the ambiguous question “Who voiced the snake in *The Jungle Book*?”. (a) The world answer support $\mathcal{W}_T(x)$ is fixed by the input, while the model-realizable support depends on the model’s knowledge. (b) Proposition 4.1 establishes a one-to-one correspondence between world-supported answers and clarification classes representing the same interpretations. (c) CIRCLECLAR estimates the model-realizable support by testing whether candidate answers are recovered through answer–clarification–answer cycles.

model that knows only one alternative, or none, is in a fundamentally different situation. We therefore ask *how to estimate the interpretations and corresponding answers a particular model knows*.

To do so, we use natural-language clarifications [12, 6], which are explicit representations of distinct interpretations of the original input. While such clarifications are widely used in prior work, their precise definition is often omitted [5, 21, 13]. Hence, we propose to characterize valid clarifications through three desiderata: (i) *uniqueness*: a clarification should admit only a single answer; (ii) *faithfulness*: that answer must also be a valid answer for the original question; and (iii) *answer equivalence*: clarifications that induce the same answer should be treated as equivalent. Our key observation is that, under these desiderata, each valid answer corresponds to exactly one clarification class, thereby providing two complementary representations of the same interpretation (Figure 1b).

This correspondence suggests a simple way to estimate what ambiguity a model can realize. If a model proposes an answer, expresses the corresponding interpretation through a clarification, and then reliably returns to the same answer under that clarification, the resulting cycle provides additional evidence that this alternative is accessible to the model. We turn this idea into CIRCLECLAR, a black-box method for estimating the model-specific set of realizable answers (Figure 1c). We evaluate CIRCLECLAR on open-domain QA using AmbigQA [12] and our curated high-quality version (AmbigQA-HQ), and on executable Text-to-SQL using AMBROSIA [14]. CIRCLECLAR achieves the best answer-set F1 in seven of twelve model–dataset settings and outperforms direct-generation baselines across all QA settings. Moreover, its cardinality can predict model-dependent ambiguity and serves as an effective basis for deciding whether to answer, clarify, or abstain. Taken together, these results support a model-dependent view of ambiguity: what matters for a particular LLM is not only which alternatives an input admits, but which of them the model can realize.

2 Related Work

Representing input ambiguity. A common way to study input ambiguity is to make the possible interpretations of an underspecified input explicit. In open-domain QA, interpretations can be expressed through disambiguated questions, either provided as annotations [12] or generated recursively as part of answering ambiguous questions [6]. In semantic parsing, where natural-language inputs are mapped to executable representations, such as SQL queries or tool calls, different interpretations can be expressed as distinct executable programs [2, 14], or first stated in natural language before being mapped to logical forms [15]. Related work also makes alternative readings explicit as structured interpretation–answer pairs [16], while Tomov et al. [20] take a distributional view, representing ambiguity through a ground-truth distribution over semantic answer classes and their relative likelihoods. Beyond such explicit representations, Kirchhof et al. [8] takes a more general view of underspecification, describing it as uncertainty about the intended task or as missing contextual information, modeled via a latent variable model.

Estimating input ambiguity. Recent work treats input ambiguity as a quantity that can be estimated from model outputs. ICE [5] treats it as a source of aleatoric uncertainty. It generates multiple clarifications of the input, obtains predictions for each, and ensembles them to separate aleatoric uncertainty from epistemic uncertainty. Spectral Uncertainty [21] extends this approach by doing soft clustering via embeddings. CLARA [13] generates multiple clarifications and uses their semantic diversity as an ambiguity signal, while Shi et al. [17] detect ambiguity from sampled-answer distributions. CLUES [25] instead uses a two-stage process from interpretations to answers, separating uncertainty across interpretations from instability in their associated answers.

Clarifications & input ambiguity. Beyond estimating ambiguity itself, other work incorporates ambiguity detection or recognition into clarification systems, benchmarks, and policies. Lee et al. [10] formulate open-domain QA as a pipeline that detects ambiguous questions, presents candidate interpretations in a clarification question, and answers based on the user’s response. CLAM [9] similarly classifies whether a question is ambiguous and selectively requests clarification before answering. CLAMBER [24] jointly benchmarks the identification of ambiguous information needs and the generation of clarification questions. INTENT-SIM [22] estimates entropy over simulated user intents to determine when clarification is beneficial. ACT [3] learns multi-turn clarification policies and evaluates implicit ambiguity recognition through models’ clarify-or-answer decisions. Zhang, Knox, and Choi [23] use simulated future conversation turns to train clarify-or-answer behavior and evaluate the resulting decisions against ambiguity labels. Su and Cardie [18] further show that ambiguity recognition capability does not necessarily translate into clarification behavior: models often identify an input as ambiguous when asked explicitly, yet default to answering in ordinary QA. Closely related to our work, BAG [1] uses sampled responses as a belief state to guide whether to answer, clarify, or abstain. Its BAG3 variant groups responses by meaning and proposes interpretations for each group to distinguish alternative readings from factual inconsistencies.

Overall, prior work studied how ambiguity can be represented, estimated, and handled through clarifications. Far less attention has been paid to how ambiguity varies with the responding model’s own knowledge. APA [7] recognizes this, defining perceived ambiguity as a property of a particular model and measuring it through the reduction in the model’s average token entropy from the original question to a self-disambiguated version. While conceptually sharing this perspective, our work goes further by formally distinguishing the two different types of ambiguity and defining not only a quantitative estimator but also actionable estimates of clarifications and their answers.

3 Why Input Ambiguity Should be Model Dependent

Ambiguity is typically treated as a property of the input: a question is ambiguous when it admits multiple valid interpretations. As such, most prior work [12, 4, 11, 6, 5, 14] studies ambiguity independently of the used model. Similar to Kim et al. [7], we propose to *treat ambiguity as dependent on model knowledge and, therefore, as a model-dependent property*. We formalize this distinction below and empirically show that it varies substantially across models (Figure 2), motivating our model-dependent framework for characterizing ambiguity.

Preliminaries. LLMs implicitly define a probability distribution $p(s)$ over arbitrary-length strings $s \in \mathcal{V}^{<\infty}$ from a vocabulary $\mathcal{V}$. Conditioned on an input $x \in \mathcal{V}^{<\infty}$, the model induces a task-specific predictive distribution $p_T(y | x)$ over answers y for a task T . We analogously denote by $p_T^*(y | x)$ the corresponding unknown ground-truth distribution. Importantly, we consider answers in a task-relevant output space $\mathcal{Y}_T$, which abstracts away distinctions that are irrelevant to the task, like surface form or syntax, by aggregating responses with similar semantic meaning into one element [19]. For question answering, for example, semantically equivalent answer strings correspond to the same element of $\mathcal{Y}_T$; for executable outputs, such as SQL queries, syntactically different forms are treated as equivalent when they return the same requested information.

World vs. Model Ambiguity. An ambiguous question x admits multiple interpretations, which may in turn lead to different correct answers. In the running example, the unspecified film yields Sterling Holloway or Scarlett Johansson. We call the set of all such answers the *world answer support*,

$$\mathcal{W}_T(x) := \text{supp } p_T^*(\cdot | x) = \{y \in \mathcal{Y}_T : p_T^*(y | x) > 0\},$$

where p_T^* denotes an oracle or ground-truth mapping. The corresponding *world ambiguity* is simply $|\mathcal{W}_T(x)|$, the number of correct answers admitted by the question. In particular, x is *world ambiguous* whenever there is more than one correct answer, i.e. $|\mathcal{W}_T(x)| > 1$.

Crucially, however, world ambiguity need not coincide with the ambiguity realized by a particular model. Returning to our example, a model that knows both Jungle Book films can realize that the question admits at least two interpretations and associate each with its corresponding answer. In contrast, another model may only know the 2016 film—due, for instance, to differences in training data, capacity, distillation, or compression—and therefore realize only Scarlett Johansson as a valid answer. The underlying question is equally ambiguous in both cases: its world answer support $\mathcal{W}_T(x)$ has not changed. What changes is which parts of this ambiguity are accessible to the model. To capture this distinction, for a model p_T we define its *model-realizable answer support* as

$$\mathcal{M}_T(x) := \text{supp } p_T(\cdot | x) \cap \mathcal{W}_T(x) = \{y \in \mathcal{W}_T(x) : p_T(y | x) > 0\}.$$
¹

Accordingly, we define the *model-realizable ambiguity* of x as $|\mathcal{M}_T(x)|$. Unlike world ambiguity, this quantity is inherently model-dependent: Two models facing the same question may realize different subsets of the valid answers. We observe precisely this effect in Figure 2: larger, more capable models tend to realize more of the ambiguity present in the world. Their model-realizable support also tracks the world support more closely across inputs, as shown by a higher input-level Spearman correlation between $|\mathcal{M}_T(x)|$ and $|\mathcal{W}_T(x)|$. Nonetheless, even the strongest model remains substantially below the mean world-support size.

This distinction between these two forms of ambiguity matters because, when deciding how a *particular model* should respond, world ambiguity alone is insufficient. Suppose $|\mathcal{W}_T(x)| > 1$, but a model realizes only one of these answers. *From an oracle perspective, the question is ambiguous. From the model’s perspective, there is no alternative interpretation on which it can meaningfully act.* Conversely, a model that realizes several valid answers should recognize that answering with any single one implicitly commits to a specific interpretation. Thus, the appropriate response—whether to answer, clarify, or abstain—depends not only on which answers exist, but on which of them are realizable by the model. Our goal is therefore not to recover $\mathcal{W}_T(x)$, but to estimate the model-specific set $\mathcal{M}_T(x)$.

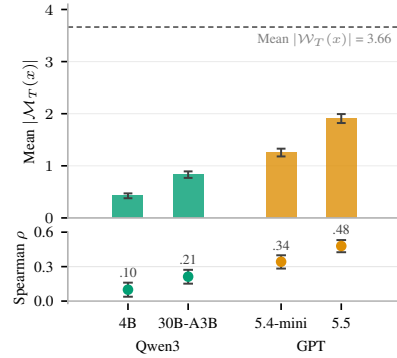


Figure 2: **Model-realizable support on AmbigQA.** Mean size (top) and Spearman correlation with world support size (bottom). Error bars show 95% bootstrap CIs over questions.

4 A Clarification Framework for Estimating Model-Dependent Ambiguity

We now turn to the question of how to estimate the model-realizable answer support $\mathcal{M}_T(x)$. To do so, we use clarifications, which provide a natural way to express interpretations as natural-language inputs for the model. While clarifications have been previously used [12, 6, 5, 13], a precise formal characterization of what constitutes a valid clarification has, to our knowledge, not been provided. This absence is consequential: without a rigorous definition, clarifications may possess undesirable properties that introduce confounds into downstream estimators. To address this gap, we are the first to formally characterize valid clarifications through a set of explicit desiderata (Section 4.1). Crucially, we prove that, under these desiderata, the set of clarifications admits an *equivalent representation* of the answer space. We then exploit this equivalence to construct CIRCLECLAR, a black-box estimator of $\mathcal{M}_T(x)$.

4.1 Formally Defining Clarifications

We begin with the most basic requirement: a clarification must actually resolve the ambiguity. Consider again the ambiguous question “*Who voiced the snake in The Jungle Book?*” in which

¹Technically, an LLM without top- p or top- k sampling assigns non-zero probability to every sequence; in practice, we consider outcomes with non-negligible probability mass.

the adaptation is unspecified. A logical clarification would be “*Who voiced the snake in the 1967 animated film The Jungle Book?*”. This reformulation resolves the ambiguity by selecting a single interpretation and, consequently, a particular answer. This motivates our first desideratum.

Desideratum 1: Uniqueness

A clarification shall uniquely determine an answer:

$$\exists y_c \in \mathcal{Y}_T \quad \text{such that} \quad p_T^*(y_c \mid c) = 1.$$

Without uniqueness, the clarification itself remains ambiguous and therefore does not isolate one particular unambiguous reading of the original input. Uniqueness alone, however, is not sufficient. For example, “*Who voiced Mowgli in the 1967 animated film The Jungle Book?*” is also unambiguous, but its answer is not a valid answer to the original question about the snake. A clarification must therefore not only resolve the ambiguity, but also preserve the meaning of the original input.

Desideratum 2: Faithfulness

The answer induced by a clarification must also be supported by the original ambiguous input prompt:

$$y_c \in \mathcal{W}_T(x).$$

Together, *Uniqueness* and *Faithfulness* ensure that a clarification selects exactly one valid answer from $\mathcal{W}_T(x)$. In our example, specifying the 1967 film selects Sterling Holloway, while specifying the 2016 film selects Scarlett Johansson; both are valid answers to the original question. Importantly, however, the clarification itself is not unique. The same interpretation can typically be expressed in different ways. For instance, “*Who voiced Kaa in Disney’s 1967 animated The Jungle Book?*” and “*Who is the voice actor for the snake Kaa in the 1967 Jungle Book film?*” differ in wording, but both select the same answer, Sterling Holloway. For our purposes, these surface-form differences are irrelevant.

Desideratum 3: Answer Equivalence

Let $\mathcal{C}_T(x)$ denote the set of valid clarifications of x for task T . Two clarifications are equivalent exactly when they induce the same answer. For two valid clarifications $c, c' \in \mathcal{C}_T(x)$,

$$c \sim_x c' \iff y_c = y_{c'}.$$

Answer Equivalence therefore groups different surface forms that express the same interpretation into *classes*. In our example, all valid clarifications that specify the 1967 film belong to one class, since they all induce the answer Sterling Holloway. We denote the class containing a clarification c by $[c]_x := \{c' \in \mathcal{C}_T(x) : c' \sim_x c\}$, and write $y_{[c]}$ for the answer shared by all clarifications in this class, while writing $\mathcal{C}_T(x)/\sim_x$ for the set of all such equivalence classes.

The three desiderata give us a useful structure: Each clarification class identifies exactly one valid answer to the original input. If we additionally assume that every valid answer can be expressed by at least one clarification, then the reverse also holds: every answer in $\mathcal{W}_T(x)$ is represented by exactly one clarification class. This gives the following correspondence (Proof in Appendix A):

Proposition 4.1 (Clarification–Answer Bijection). *Suppose clarifications satisfy D1–D3 and every $y \in \mathcal{W}_T(x)$ admits at least one valid clarification. Then the map from clarification to response*

$$\Phi_x : \mathcal{C}_T(x)/\sim_x \longrightarrow \mathcal{W}_T(x), \quad \Phi_x([c]_x) = y_{[c]},$$

is a bijection.

For each $y \in \mathcal{W}_T(x)$, we write $[c_y]_x$ for the unique clarification class corresponding to y under this bijection. Returning once more to our example, the interpretation referring to the 1967 film can now be represented either by the answer *Sterling Holloway* or by the entire class of clarifications that uniquely lead to this answer, such as “*Who voiced the snake in the 1967 animated film The Jungle Book?*”. More generally, Proposition 4.1 gives us two equivalent representations of each interpretation: its answer y and a class $[c_y]_x$ of natural-language clarifications that induce it. Hence, starting from an answer y , we can move to its corresponding clarification class $[c_y]_x$, and any clarification from this class maps back to the same answer y . This yields the cycle $y \longrightarrow [c_y]_x \longrightarrow y$, the key consistency relation underlying the estimator introduced next.

Algorithm 1 CircleClar

Require: Input x , model M , proposal budget B , recovery samples R , stability threshold τ

Ensure: Estimated model-realizable answer support $\widehat{\mathcal{M}}_T(x)$

```
1:  $\widehat{\mathcal{M}}_T(x) \leftarrow \emptyset$ 
2: Draw  $B$  answer collections  $Y_1, \dots, Y_B$  from  $M$  on  $x$ 
3:  $\widetilde{\mathcal{Y}}_T(x) \leftarrow \text{Canon}_T\left(\bigcup_{b=1}^B Y_b\right)$ 
4: for  $\widetilde{y} \in \widetilde{\mathcal{Y}}_T(x)$  do
5:   Generate a clarification  $c_{\widetilde{y}}$  targeting  $\widetilde{y}$ 
6:   Re-answer  $c_{\widetilde{y}}$  independently  $R$  times and map the responses to semantic answer classes
      $z_1, \dots, z_R$ 
7:    $\widehat{q}_{\widetilde{y}} \leftarrow \frac{1}{R} \sum_{r=1}^R \mathbb{1}\{z_r = \widetilde{y}\}$ 
8:   if  $\widetilde{y}$  is the unique most frequent answer and  $\widehat{q}_{\widetilde{y}} \geq \tau$  then
9:      $\widehat{\mathcal{M}}_T(x) \leftarrow \widehat{\mathcal{M}}_T(x) \cup \{\widetilde{y}\}$ 
10:  end if
11: end for
12: return  $\widehat{\mathcal{M}}_T(x)$ 
```

4.2 CIRCLECLAR: Estimating Model-Realizable Answer Support

The answer–clarification correspondence established under the ideal task distribution p_T^* provides a consistency criterion for testing candidate answers using the model distribution p_T : if a candidate $\widetilde{y}$ corresponds to an answer of a valid interpretation of x , then expressing that interpretation through a clarification should make $\widetilde{y}$ uniquely correct, and the model should recover it when answering the clarified input. Reliably recovering $\widetilde{y}$ under p_T therefore provides evidence that $\widetilde{y} \in \mathcal{M}_T(x)$.

As illustrated in Figure 1c, CIRCLECLAR operationalizes this idea in two steps: it first constructs a set of candidate outcomes supported by the model, and then uses the answer–clarification–answer cycle as a consistency test to filter these candidates. Concretely, we first query the model B times with an enumeration prompt that asks for all plausible answers to the original input x , obtaining answer collections $Y_1, \dots, Y_B$. We then take the union over these B answer sets and map the resulting responses to semantic answer classes, yielding the candidate set $\widetilde{\mathcal{Y}}_T(x)$. This candidate set provides an approximation to the model’s predictive support, $\widetilde{\mathcal{Y}}_T(x) \approx \text{supp } p_T(\cdot \mid x)$. For each candidate $\widetilde{y} \in \widetilde{\mathcal{Y}}_T(x)$, we ask the model to generate a clarification $c_{\widetilde{y}}$ that makes the corresponding interpretation explicit. We then close the cycle by prompting the same LLM to answer this clarification again. If the clarification truly represents the same interpretation, the model should return to the original candidate $\widetilde{y}$. We capture this through the recovery probability $p_T(\widetilde{y} \mid c_{\widetilde{y}})$, retaining candidates whose recovery probability meets or exceeds a threshold τ . In practice, we approximate $p_T(\widetilde{y} \mid c_{\widetilde{y}})$ using R Monte Carlo samples from $p_T(\cdot \mid c_{\widetilde{y}})$. The complete algorithm is in Algorithm 1 with prompts in Appendix E.

5 Experiments

Our primary evaluation asks how accurately CIRCLECLAR recovers the model-realizable answer support. We then evaluate two downstream uses of the recovered support in QA: using its cardinality to detect model-dependent ambiguity and deriving the response action from the estimated support.

Datasets. We evaluate CIRCLECLAR across two domains. For open-domain question answering, we use 1,000 ambiguous questions from AmbigQA [12], together with AmbigQA-HQ, a manually curated set of 100 questions derived primarily from AmbigQA with more complete and reliable rewrite–answer annotations (see Appendix B). For executable Text-to-SQL, we evaluate on 96 ambiguous requests from AMBROSIA [14]. Since we treat final answers y by their task-relevant semantic meaning rather than as raw strings, we construct the corresponding equivalence classes using an LLM-based semantic grouping procedure tailored to the task (Appendix E).

Model-Realizable Reference Support. We construct $\mathcal{W}_T(x)$ from the distinct reference answers associated with the dataset-provided resolved inputs. To construct the model-specific reference

Table 1: **Model-realizable answer support recovery on QA and Text-to-SQL.** We report mean per-question set $F_1 \pm$ standard deviation and exact set match (EM). **Best** and **second-best** results are highlighted.

Dataset	Method	Qwen3-4B F_1 / EM	Qwen3-30B-A3B F_1 / EM	GPT-5.4-mini F_1 / EM	GPT-5.5 F_1 / EM
AmbigQA HQ	ICE	39.23 ± 46.09 / 33.00	52.52 ± 41.97 / 33.00	63.93 ± 34.35 / 31.00	77.91 ± 29.61 / 50.00
	Sampling-50	33.07 ± 43.78 / 25.00	43.63 ± 40.11 / 22.00	55.51 ± 35.23 / 24.00	61.79 ± 27.13 / 21.00
	1-Enumeration	23.42 ± 38.45 / 15.00	41.38 ± 36.37 / 12.00	59.40 ± 34.74 / 28.00	68.48 ± 29.52 / 35.00
	CIRCLECLAR	60.02 ± 46.26 / 52.00	64.88 ± 40.76 / 49.00	74.09 ± 34.78 / 54.00	68.85 ± 31.59 / 35.00
AmbigQA	ICE	33.71 ± 44.54 / 26.90	41.04 ± 42.63 / 25.50	55.18 ± 42.50 / 35.50	63.93 ± 35.82 / 31.70
	Sampling-50	24.85 ± 40.58 / 19.60	31.66 ± 42.25 / 22.30	39.56 ± 40.10 / 20.60	44.84 ± 36.69 / 17.80
	1-Enumeration	14.33 ± 32.46 / 10.20	24.50 ± 33.90 / 7.40	38.28 ± 36.54 / 12.30	46.93 ± 35.57 / 16.80
	CIRCLECLAR	48.28 ± 48.53 / 44.70	46.26 ± 45.28 / 35.00	46.55 ± 41.90 / 27.60	49.32 ± 36.49 / 19.90
AMBROSIA	ICE	44.44 ± 37.49 / 17.71	36.41 ± 33.73 / 8.33	41.84 ± 32.92 / 8.33	48.02 ± 29.95 / 8.33
	Sampling-50	34.44 ± 40.46 / 17.71	39.50 ± 36.61 / 11.46	36.13 ± 31.34 / 5.21	45.51 ± 31.33 / 7.29
	1-Enumeration	38.70 ± 34.62 / 11.46	45.45 ± 36.91 / 18.75	53.26 ± 37.99 / 28.12	51.98 ± 32.31 / 11.46
	CIRCLECLAR	42.64 ± 40.67 / 21.88	51.39 ± 39.76 / 25.00	56.39 ± 37.84 / 26.04	48.63 ± 34.53 / 12.50

$\mathcal{M}_T(x)$, we test whether the evaluated model reliably recovers each reference answer y from its associated ground-truth clarification c_y and if so, include it. (See Appendix E for more details).

5.1 Estimating the Model-Realizable Answer Support $\mathcal{M}_T(x)$

Baselines. Existing methods cannot directly recover a model-realizable answer set. We therefore adapt ICE [5] to recover answer sets by retaining answers with probability above a threshold, oracle-tuned per model–dataset pair to maximize the mean set F1. CIRCLECLAR instead uses the fixed threshold $\tau = 0.6$, with enumeration budgets $B = 6$ for QA and $B = 5$ for Text-to-SQL. We further compare against two direct-generation baselines: SAMPLING-50, which samples 50 times, and 1-ENUMERATION, which asks the model once to enumerate all plausible answers. Full details are provided in Appendix E.

Results. Table 1 shows that CIRCLECLAR performs strongly across datasets, achieving particularly large gains over baselines on smaller models and on the curated AmbigQA-HQ benchmark. The overall performance gap between AmbigQA and AmbigQA-HQ likely stems from annotation errors in the standard AmbigQA, which introduce noisy estimates of the ground-truth set. Across all methods, performance gaps shrink on larger models. Notably, the simple baseline of enumerating possible answers one-shot achieves remarkably high performance on larger models, indicating that increased scale enables models to surface multiple plausible alternatives directly within a single structured response. Furthermore, precision and recall breakdowns in Appendix C.1 show that CIRCLECLAR’s gains are most consistent in precision. Moreover, evaluations on unambiguous, single-answer questions in Appendix C.3 confirm that our method largely avoids hallucinating non-existent alternatives in unambiguous cases and, in fact, significantly outperforms other methods on smaller models. Finally, ablations over the candidate proposal budget in Appendix C.2 reveal that expanding the candidate size decreases performance for smaller models but improves performance for larger models. This contrast may indicate that smaller models are primarily constrained by precision, whereas larger models are constrained by recall.

Effect of the consistency check. To isolate the effect of the consistency check, we compare the unfiltered candidate set $\tilde{\mathcal{Y}}_T(x)$ with the final estimate $\widehat{\mathcal{M}}_T(x)$ obtained after consistency filtering. As shown in Figure 3, the consistency check yields substantial gains in precision across nearly all settings. Because this filtering step can only remove candidates, recall naturally decreases, but the resulting trade-off is generally small compared to the sharp increase in precision. Notably, the magnitude of the precision gain diminishes with model scale, suggesting that larger models hallucinate fewer invalid initial candidates, and explaining why the simple one-shot enumeration becomes increasingly effective for larger models.

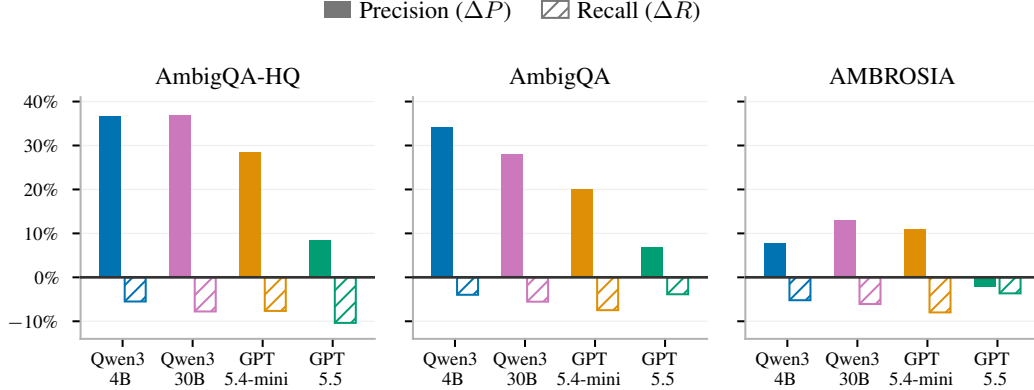


Figure 3: **Effect of the consistency check on support recovery.** Change in macro precision and recall from the initial candidate set $\tilde{\mathcal{Y}}_T(x)$ to the final CIRCLECLAR estimate $\hat{\mathcal{M}}_T(x)$. We report $\Delta P = P(\hat{\mathcal{M}}_T) - P(\tilde{\mathcal{Y}}_T)$ and analogously for ΔR .

5.2 Downstream Uses of Model-Realizable Support

As discussed in Section 3, the same ambiguous input should lead to different behavior from different models, depending on the interpretations they can actually resolve. This motivates using the recovered model-realizable support not only as an object of estimation, but as a basis for deciding whether the model faces unresolved ambiguity and how it should respond. We examine this through model-dependent ambiguity detection and response selection of whether to abstain, answer, or clarify.

Ambiguity detection. Unlike conventional ambiguity evaluation, which asks whether the input itself admits multiple valid answers, i.e., whether $|\mathcal{W}_T(x)| \geq 2$, we label an input as ambiguous for model M when $|\mathcal{M}_T(x)| \geq 2$. As such, the cardinality of the estimated model realizable answer support provides a direct ambiguity score, i.e. $|\hat{\mathcal{M}}_T(x)|$. We compare against ICE [5], which uses mutual information between clarifications and answers, CLARA-D [13], which measures dispersion among generated clarifications, INTENT-SIM [22], which estimates entropy over simulated intents, and a modified APA [7] version $APA^\dagger$, which computes information gain via semantic entropy of the original vs self-disambiguated question. We evaluate all ambiguity scores using AUROC.

Figure 4a demonstrates that the cardinality of the recovered candidate set effectively identifies whether model-dependent ambiguity is present. CIRCLECLAR achieves the highest overall performance, with its most pronounced advantage occurring on AmbigQA-HQ. This improvement reflects again the higher quality of AmbigQA-HQ’s reference sets: annotations in standard AmbigQA are frequently incomplete or noisy, which artificially distorts the true reference cardinality and degrades the quality of the resulting ambiguity labels (Appendix B). Notably, baseline methods perform close to random guessing. While prior work shows that these baselines perform well when evaluated against $|\mathcal{W}_T(x)|$, our findings highlight again the crucial distinction: optimizing for $|\mathcal{W}_T(x)|$ does not translate to detecting whether the model itself perceives a question as ambiguous.

Response policy. The question of whether to abstain, answer, or clarify follows naturally from the model-realizable support. If the support is empty, the model has no valid answer it can provide (ABSTAIN); if it contains a single answer, there is no alternative for the model to resolve (ANSWER); and if it contains multiple answers, the appropriate choice depends on the intended interpretation (CLARIFY). The ground-truth label is hence obtained following this logic on $\mathcal{M}_T(x)$ while the predictions for CIRCLECLAR are based on $\hat{\mathcal{M}}_T(x)$. To assess whether explicitly estimating $\hat{\mathcal{M}}_T(x)$ is beneficial, we compare against two one-shot policies that operate directly on the original input: *Direct (Generic)*, based on general ambiguity and answerability criteria, and *Direct (Model-Dependent)*, which applies the same model-dependent rule without constructing an answer set. We evaluate all three methods using balanced accuracy over the potential responses. Prompts are provided in Appendix E.3.

Figure 4b shows that deriving the response action from the recovered support is generally more reliable than predicting it directly. CIRCLECLAR performs best overall, while explicitly stating the model-

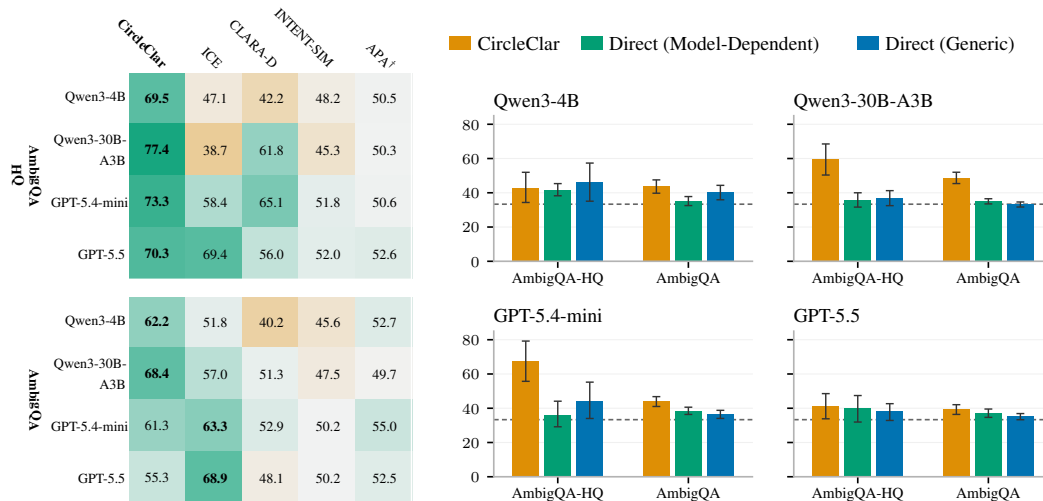


Figure 4a: Ambiguity detection (AU-ROC). CIRCLECLAR is strongest overall; its clearest gains occur on AmbigQA-HQ.

Figure 4b: Three-way policy (BAcc). CIRCLECLAR leads in most settings when deciding whether to abstain, clarify, or answer. Error bars 95% bootstrap CIs; dashed line is constant-policy.

dependent decision rule provides no consistent advantage over *Direct (Generic)*, supporting the value of explicitly recovering the answer support. Appendix D complements this analysis by evaluating the set-estimating methods in Table 1 under progressively stricter answer-content requirements.

6 Discussion

6.1 Limitations

CIRCLECLAR is a precision-oriented estimator whose recall is limited by the initial candidate set: interpretations that the model can answer when explicitly specified may still be unlikely to arise during candidate generation and therefore remain undiscovered. Relatedly, we focus on the *support* of model-realizable answers rather than the relative likelihood of interpretations, treating recovered alternatives equally through a set representation and its cardinality. Further, successful recovery through an answer–clarification–answer cycle provides evidence of model realizability but does not guarantee correctness, as a model may consistently reproduce a factual error; under black-box access, such confidently held errors are generally difficult to distinguish from correct knowledge. Our evaluation is also limited by the quality and coverage of ambiguity annotations, which may omit valid interpretations or introduce unsupported ones. We mitigate this by constructing the curated AmbigQA-HQ benchmark (Appendix B), but cannot guarantee complete coverage. Finally, $\mathcal{M}_T(x)$ captures *answerable* model-dependent ambiguity and does not include interpretations that a model can recognize but cannot answer, nor purely linguistic ambiguity it may detect without resolving. Explicitly separating interpretation recognition from answerability is an interesting direction for future work.

6.2 A Model-Dependent View of Ambiguity

Our results support a view of ambiguity that distinguishes what an input permits from what a particular model can realize. While the world may admit many valid interpretations, only a subset may be accessible within a model’s knowledge, and it is this model-realizable ambiguity that determines the alternatives on which the model can meaningfully act. This distinction complements the conventional, world-centric view of ambiguity and provides a more model-specific account of how underspecified inputs are represented and handled. Our clarification framework provides a natural way to characterize this object by establishing a one-to-one correspondence between valid answers and equivalence classes of clarifications. CIRCLECLAR is one concrete way of exploiting this correspondence under black-box access, but the underlying perspective is more general. We hope this view provides a useful basis for future work on ambiguity-aware language models and their interaction with users.

References

- [1] Joris Baan, Wilker Aziz, Barbara Plank, and Raquel Fernández. Clarify, abstain or answer? strategising in conversation with belief-augmented generation. *arXiv preprint arXiv:2605.25831*, 2026.
- [2] Adithya Bhaskar, Tushar Tomar, Ashutosh Sathe, and Sunita Sarawagi. Benchmarking and improving text-to-SQL generation under ambiguity. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7053–7074, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.436. URL <https://aclanthology.org/2023.emnlp-main.436/>.
- [3] Maximillian Chen, Ruoxi Sun, Tomas Pfister, and Sercan O Arik. Learning to clarify: Multi-turn conversations with action-based contrastive self-training. In *The Thirteenth International Conference on Learning Representations*, 2025. URL <https://openreview.net/forum?id=SIE6VFps9x>.
- [4] Jeremy Cole, Michael Zhang, Daniel Gillick, Julian Eisenschlos, Bhuwan Dhingra, and Jacob Eisenstein. Selectively answering ambiguous questions. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 530–543, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.35. URL <https://aclanthology.org/2023.emnlp-main.35/>.
- [5] Bairu Hou, Yujian Liu, Kaizhi Qian, Jacob Andreas, Shiyu Chang, and Yang Zhang. Decomposing uncertainty for large language models through input clarification ensembling. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 19023–19042. PMLR, 21–27 Jul 2024. URL <https://proceedings.mlr.press/v235/hou24b.html>.
- [6] Gangwoo Kim, Sungdong Kim, Byeongguk Jeon, Joonsuk Park, and Jaewoo Kang. Tree of clarifications: Answering ambiguous questions with retrieval-augmented large language models. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 996–1009, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.63. URL <https://aclanthology.org/2023.emnlp-main.63/>.
- [7] Hyuhng Joon Kim, Youna Kim, Cheonbok Park, Junyeob Kim, Choonghyun Park, Kang Min Yoo, Sang-goo Lee, and Taeuk Kim. Aligning language models to explicitly handle ambiguity. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1989–2007, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.119. URL <https://aclanthology.org/2024.emnlp-main.119/>.
- [8] Michael Kirchhof, Gjergji Kasneci, and Enkelejda Kasneci. Position: Uncertainty quantification needs reassessment for large language model agents. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu, editors, *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 81665–81677. PMLR, 13–19 Jul 2025. URL <https://proceedings.mlr.press/v267/kirchhof25b.html>.
- [9] Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Clam: Selective clarification for ambiguous questions with generative language models, 2023. URL <https://arxiv.org/abs/2212.07769>.
- [10] Dongryeol Lee, Segwang Kim, Minwoo Lee, Hwanhee Lee, Joonsuk Park, Sang-Woo Lee, and Kyomin Jung. Asking clarification questions to handle ambiguity in open-domain QA. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 11526–11544, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.772. URL <https://aclanthology.org/2023.findings-emnlp.772/>.

- [11] Alisa Liu, Zhaofeng Wu, Julian Michael, Alane Suhr, Peter West, Alexander Koller, Swabha Swayamdipta, Noah Smith, and Yejin Choi. We’re afraid language models aren’t modeling ambiguity. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 790–807, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.51. URL <https://aclanthology.org/2023.emnlp-main.51/>.
- [12] Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. AmbigQA: Answering ambiguous open-domain questions. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5783–5797, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.466. URL <https://aclanthology.org/2020.emnlp-main.466/>.
- [13] Félix Rosseeuw, Adil Bahaj, and Mounir Ghogho. CLARA: Clarification-driven measurement of input ambiguity in LLMs, 2025. URL <https://openreview.net/forum?id=GAh52Zncfk>.
- [14] Irina Saparina and Mirella Lapata. Ambrosia: A benchmark for parsing ambiguous questions into database queries. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 90600–90628. Curran Associates, Inc., 2024. doi: 10.52202/079017-2876. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/a4c942a8405cc910f0a833d28d2573cc-Paper-Datasets_and_Benchmarks_Track.pdf.
- [15] Irina Saparina and Mirella Lapata. Disambiguate first, parse later: Generating interpretations for ambiguity resolution in semantic parsing. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Findings of the Association for Computational Linguistics: ACL 2025*, pages 16825–16839, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.863. URL <https://aclanthology.org/2025.findings-acl.863/>.
- [16] Irina Saparina and Mirella Lapata. Reasoning about intent for ambiguous requests, 2026. URL <https://arxiv.org/abs/2511.10453>.
- [17] Zhengyan Shi, Giuseppe Castellucci, Simone Filice, Saar Kuzi, Elad Kravi, Eugene Agichtein, Oleg Rokhlenko, and Shervin Malmasi. Ambiguity detection and uncertainty calibration for question answering with large language models. In Trista Cao, Anubrata Das, Tharindu Kumarage, Yixin Wan, Satyapriya Krishna, Ninareh Mehrabi, Jwala Dhamala, Anil Ramakrishna, Aram Galystan, Anoop Kumar, Rahul Gupta, and Kai-Wei Chang, editors, *Proceedings of the 5th Workshop on Trustworthy NLP (TrustNLP 2025)*, pages 41–55, Albuquerque, New Mexico, May 2025. Association for Computational Linguistics. ISBN 979-8-89176-233-6. doi: 10.18653/v1/2025.trustnlp-main.4. URL <https://aclanthology.org/2025.trustnlp-main.4/>.
- [18] Jinyan Su and Claire Cardie. Knowing but not showing: LLMs recognize ambiguity but rarely ask clarifying questions, 2026. URL <https://arxiv.org/abs/2605.25284>.
- [19] Tim Tomov, Dominik Fuchsgruber, and Stephan Günnemann. Task-awareness improves LLM generations and uncertainty. In *Forty-third International Conference on Machine Learning*, 2026. URL <https://openreview.net/forum?id=Zuews4ULW6>.
- [20] Tim Tomov, Dominik Fuchsgruber, Tom Wollschläger, and Stephan Günnemann. The illusion of certainty: Uncertainty quantification for LLMs fails under ambiguity, 2026. URL <https://arxiv.org/abs/2511.04418>.
- [21] Nassim Walha, Sebastian Gregor Gruber, Thomas Decker, Yinchong Yang, Alireza Javanmardi, Eyke Hüllermeier, and Florian Buettner. Fine-grained uncertainty decomposition in large language models: A spectral approach. In *The 40th Annual AAAI Conference on Artificial Intelligence*, number 31, pages 26090–26098, 2026. URL <http://nbn-resolving.de/urn/resolver.pl?urn=nbn:de:bvb:19-epub-135635-1>.

- [22] Michael JQ Zhang and Eunsol Choi. Clarify when necessary: Resolving ambiguity through interaction with LMs. In Luis Chiruzzo, Alan Ritter, and Lu Wang, editors, *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 5541–5558, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-195-7. doi: 10.18653/v1/2025.findings-naacl.306. URL <https://aclanthology.org/2025.findings-naacl.306/>.
- [23] Michael JQ Zhang, W. Bradley Knox, and Eunsol Choi. Modeling future conversation turns to teach LLMs to ask clarifying questions. In *The Thirteenth International Conference on Learning Representations*, 2025. URL <https://openreview.net/forum?id=cwuSAR7EKd>.
- [24] Tong Zhang, Peixin Qin, Yang Deng, Chen Huang, Wenqiang Lei, Junhong Liu, Dingnan Jin, Hongru Liang, and Tat-Seng Chua. CLAMBER: A benchmark of identifying and clarifying ambiguous information needs in large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10746–10766, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.578. URL <https://aclanthology.org/2024.acl-long.578/>.
- [25] Angelo Ziletti and Leonardo D’Ambrosi. Disentangling ambiguity from instability in large language models: A clinical text-to-SQL case study. In Asma Ben Abacha, Steven Bethard, Danielle Bitterman, Tristan Naumann, and Kirk Roberts, editors, *Proceedings of the 8th Workshop on Clinical Natural Language Processing (Clinical NLP) @ LREC 2026*, pages 369–380, Palma, Mallorca (Spain), May 2026. ELRA Language Resources Association (ELRA). doi: 10.63317/4qse3ioqyh8s. URL <https://aclanthology.org/2026.clinicalnlp-1.39/>.

A Proof of Proposition 4.1

Proposition 4.1 (Clarification–Answer Bijection). *Suppose clarifications satisfy D1–D3 and every $y \in \mathcal{W}_T(x)$ admits at least one valid clarification. Then the map from clarification to response*

$$\Phi_x : \mathcal{C}_T(x) / \sim_x \longrightarrow \mathcal{W}_T(x), \quad \Phi_x([c]_x) = y_{[c]},$$

is a bijection.

Proof. By D1, every valid clarification $c \in \mathcal{C}_T(x)$ induces a unique answer y_c , and by D2, $y_c \in \mathcal{W}_T(x)$. By D3, all clarifications in the same equivalence class induce the same answer, so Φ_x is well-defined.

To show injectivity, suppose

$$\Phi_x([c]_x) = \Phi_x([c']_x).$$

Then $y_c = y_{c'}$, and D3 implies $c \sim_x c'$. Hence $[c]_x = [c']_x$.

To show surjectivity, let $y \in \mathcal{W}_T(x)$. By assumption, there exists a valid clarification $c \in \mathcal{C}_T(x)$ that induces y . Therefore,

$$\Phi_x([c]_x) = y.$$

Thus, Φ_x is both injective and surjective, and hence is a bijection. $\square$

B Construction of AmbigQA-HQ

Our evaluation requires a reliable and sufficiently complete reference set of valid answers for each ambiguous question. AmbigQA’s disambiguated rewrites and answers [12] provide a natural empirical reference for $\mathcal{W}_T(x)$, from which we construct each model-specific $\mathcal{M}_T(x)$. However, the original annotations do not always meet this requirement. Some questions labeled ambiguous support only one answer under their original wording; their additional answers are obtained only by changing the question. For genuinely ambiguous questions, valid readings may be missing, a rewrite may remain unresolved or introduce an unsupported reading, several rewrites may yield the same answer, or an answer may be factually incorrect. These errors change both the membership and the size of the reference answer set. We therefore construct AmbigQA-HQ primarily by selecting genuinely ambiguous questions from AmbigQA and then verifying, repairing, and completing their disambiguated rewrites and answers. Of the 100 questions, 96 were selected from AmbigQA and four were newly constructed to cover additional ambiguity patterns; their answers were verified against authoritative sources.

Question selection. For each AmbigQA question, we ask whether its original wording genuinely supports more than one answer. We include it only when $|\mathcal{W}_T(x)| > 1$. We exclude cases in which the alternatives come from rewrites that change what the question asks. Table 2 shows both kinds of cases. The excluded rewrites either split a multi-part request or introduce a restriction absent from the original question.

Curating rewrites and answers. For every question, we manually examine its disambiguated rewrites and answers. Each rewrite should isolate one reading of the original question, and its answer should be factually correct. Guided by D1–D3, we remove unsupported rewrites, split rewrites that still allow several answers, and merge those that lead to the same answer. We also add missing plausible readings and make small wording changes where needed. This gives us a more reliable and complete reference answer set for evaluation. Table 3 shows four representative examples.

Table 2: Examples of question selection.

Original question	Original AmbigQA rewrites	Decision
“When did the Ottoman Empire take over Palestine?”	“When did the ottoman empire take over palestine for the first time ?” “When did the ottoman empire take over palestine for the second time ?”	Included: <i>take over</i> can refer to either of two historical events.
“Is red bell pepper a fruit or vegetable?”	“In terms of botanicals , is a red bell pepper a fruit or vegetables?” “In culinary , is a red bell pepper a fruit or vegetable?”	Included: botanical and culinary classifications give different valid answers.
“When did the Winter Olympic Games begin and where?”	“ When did the winter olympic games begin?” “ Where did the first winter olympic games begin?”	Not ambiguous: the rewrites split one two-part request.
“What is the latest version of iPhone software?”	“What is the latest version of iphone software, released in 2017 ?” “What is the latest version of iphone software, released in 2016 ?” “What is the latest version of iphone software, released in June 2014 ?”	Not ambiguous: <i>latest</i> has one answer; the years change its scope.
“Who landed the first quad jump in figure skating?”	“Who was the first person to land a quad jump in figure skating?” “Who was the first woman to land a quad jump in figure skating?” “Who landed the first quad jump in figure skating ever ?”	Not ambiguous: <i>first</i> has one answer; “woman” changes its scope.
“Who has the highest number of likes on Instagram?”	“Who has the highest number of likes on instagram in 2017 ?” “Who has the highest number of likes on instagram in 2016 ?” “Who has the highest number of likes on instagram in 2015 ?”	Not ambiguous: <i>highest</i> has one answer; the years change its scope.

Table 3: Examples of rewrite and answer curation in AmbigQA-HQ.

Original AmbigQA	Curated AmbigQA-HQ
D1 (Uniqueness): Split a multi-answer rewrite Question: “Who is the girl in ZZ Top’s ‘Legs’ video?”	
Salesgirl → Wendy Frazier Eliminator girls → {Danièle Arnaud, Jeana Tomasino, Kymberly Herrin}	Shoe-store clerk → Wendy Frazier Dark-haired girl → Jeana Tomasino Girl wearing black → Danièle Arnaud Girl wearing red → Kymberly Herrin
D2 (Faithfulness): Remove an unsupported rewrite Question: “When did ‘The Haves and the Have Nots’ start?”	
TV series → May 28, 2013 Stage play → 2011 Episodes 105–107 → three dates in 2017	TV series → May 28, 2013 Stage play → 2011
D3 (Answer equivalence): Merge duplicate rewrites Question: “Who made the song ‘Total Eclipse of the Heart’?”	
Wrote/produced → Jim Steinman (four rewrites) Originally recorded → Bonnie Tyler (two rewrites)	Wrote and produced → Jim Steinman Originally recorded → Bonnie Tyler
Factual correction Question: “When was the first ‘Texas Chainsaw Massacre’ movie made?”	
Filmed → July 1974	Filmed → July–August 1973

C Additional Answer-Set Recovery Results

C.1 Precision and Recall for Model-Realizable Answer Support

Table 4 complements the macro set F_1 and exact set match results in Table 1 with macro precision and recall. For each input x we compute

$$P(x) = \frac{|\widehat{\mathcal{M}}_T(x) \cap \mathcal{M}_T(x)|}{|\widehat{\mathcal{M}}_T(x)|}, \quad R(x) = \frac{|\widehat{\mathcal{M}}_T(x) \cap \mathcal{M}_T(x)|}{|\mathcal{M}_T(x)|}.$$

If the estimated support is empty, precision is one only when the reference support is also empty, and zero otherwise. Recall is one whenever the reference support is empty. All other evaluation choices match Table 1. Overall, CIRCLECLAR shows a more consistent advantage in precision than in recall.

C.2 Effect of Repeated Enumeration and Consistency Filtering

The gains from CIRCLECLAR could in principle come only from generating more candidate answers, rather than from checking those candidates through answer-clarification-answer cycles. We therefore examine both how the complete method changes with the enumeration budget and how its consistency cycles change the candidate set. For each $k \in \{1, \dots, 6\}$, we construct

$$\tilde{\mathcal{Y}}_{T,k}(x) = \text{Canon}_T \left(\bigcup_{b=1}^k Y_b \right).$$

For panel (a), we set $B = k$, apply the consistency cycles at every budget, and obtain $\widehat{\mathcal{M}}_{T,B}(x)$. We report macro set $F_1(\widehat{\mathcal{M}}_{T,B}(x), \mathcal{M}_T(x))$. Panels (b) and (c) use a different horizontal axis: the points $1, \dots, 5$ report $\tilde{\mathcal{Y}}_{T,k}(x)$ for $k = 1, \dots, 5$, $\tilde{\mathcal{Y}}_T(x)$ denotes the candidate set at $k = 6$, and $\widehat{\mathcal{M}}_T(x)$ denotes the final set after applying the consistency cycles.

Panel (a) of Figure 5 shows that the effect of B on the complete method is model- and dataset-dependent rather than uniformly increasing. On AmbigQA-HQ, performance increases for the GPT models and decreases for the Qwen models; on AmbigQA, it decreases for all models except GPT-5.5, which remains nearly constant. Panels (b) and (c), by contrast, show that as the pre-consistency candidate set accumulates from $k = 1$ to $k = 6$, macro precision mostly decreases or remains nearly flat, whereas macro recall increases or saturates. The final transition from $\tilde{\mathcal{Y}}_T(x)$ to $\widehat{\mathcal{M}}_T(x)$ reverses this trade-off: the consistency cycles increase macro precision across both datasets and all four models at the cost of lower macro recall.

C.3 Answer-Set Recovery on Single-Answer Questions

We additionally evaluate all methods on 100 unambiguous questions from AmbigQA, for which the model-realizable answer support is empty or a singleton. As shown in Table 5, CIRCLECLAR achieves the highest set F1 for three of the four models. For GPT-5.5, all methods perform similarly, with Sampling-50 slightly strongest.

Table 4: Macro precision and recall (%) corresponding to the set- F_1 results in Table 1. Each cell reports P/R ; recall is one when the reference support is empty. Green and orange mark the best and second-best method, respectively.

Dataset	Method	Qwen3-4B P/R	Qwen3-30B-A3B P/R	GPT-5.4-mini P/R	GPT-5.5 P/R
AmbigQA HQ	ICE	39.83 / 83.58	54.58 / 74.67	69.08 / 75.77	78.50 / 84.75
	Sampling-50	37.00 / 81.58	54.37 / 69.15	66.55 / 67.40	84.19 / 58.02
	1-Enumeration	26.34 / 84.58	39.92 / 79.98	60.25 / 74.57	79.15 / 68.62
	CIRCLECLAR	63.50 / 79.58	72.83 / 73.72	81.93 / 75.48	80.90 / 67.58
AmbigQA	ICE	33.39 / 87.51	39.86 / 84.58	56.78 / 77.83	61.58 / 83.11
	Sampling-50	25.59 / 85.07	35.11 / 74.55	44.82 / 69.61	59.83 / 52.78
	1-Enumeration	15.56 / 84.95	22.52 / 78.77	37.13 / 74.81	54.91 / 61.14
	CIRCLECLAR	49.49 / 81.78	48.35 / 74.72	50.68 / 71.13	58.22 / 61.21
AMBROSIA	ICE	53.91 / 47.22	44.44 / 52.78	50.09 / 51.56	47.40 / 62.15
	Sampling-50	42.71 / 35.42	52.29 / 51.56	43.05 / 49.31	59.15 / 48.96
	1-Enumeration	43.23 / 52.95	48.96 / 65.45	56.77 / 66.32	65.45 / 55.73
	CIRCLECLAR	51.74 / 43.06	59.38 / 60.94	60.28 / 67.53	58.37 / 53.65

Table 5: Macro set F_1 and exact set match (EM) (%) for recovering $\mathcal{M}_T(x)$ on 100 unambiguous AmbigQA questions. Green and orange mark the best and second-best method, respectively.

Dataset	Method	Qwen3-4B F_1 / EM	Qwen3-30B-A3B F_1 / EM	GPT-5.4-mini F_1 / EM	GPT-5.5 F_1 / EM
AmbigQA Single-answer	ICE	43.90 / 37.00	61.32 / 54.00	71.20 / 59.00	88.50 / 86.00
	Sampling-50	54.22 / 53.00	69.17 / 68.00	78.83 / 77.00	88.67 / 86.00
	1-Enumeration	48.67 / 48.00	57.40 / 40.00	71.83 / 59.00	87.67 / 84.00
	CIRCLECLAR	68.67 / 68.00	70.07 / 67.00	82.00 / 80.00	87.17 / 84.00

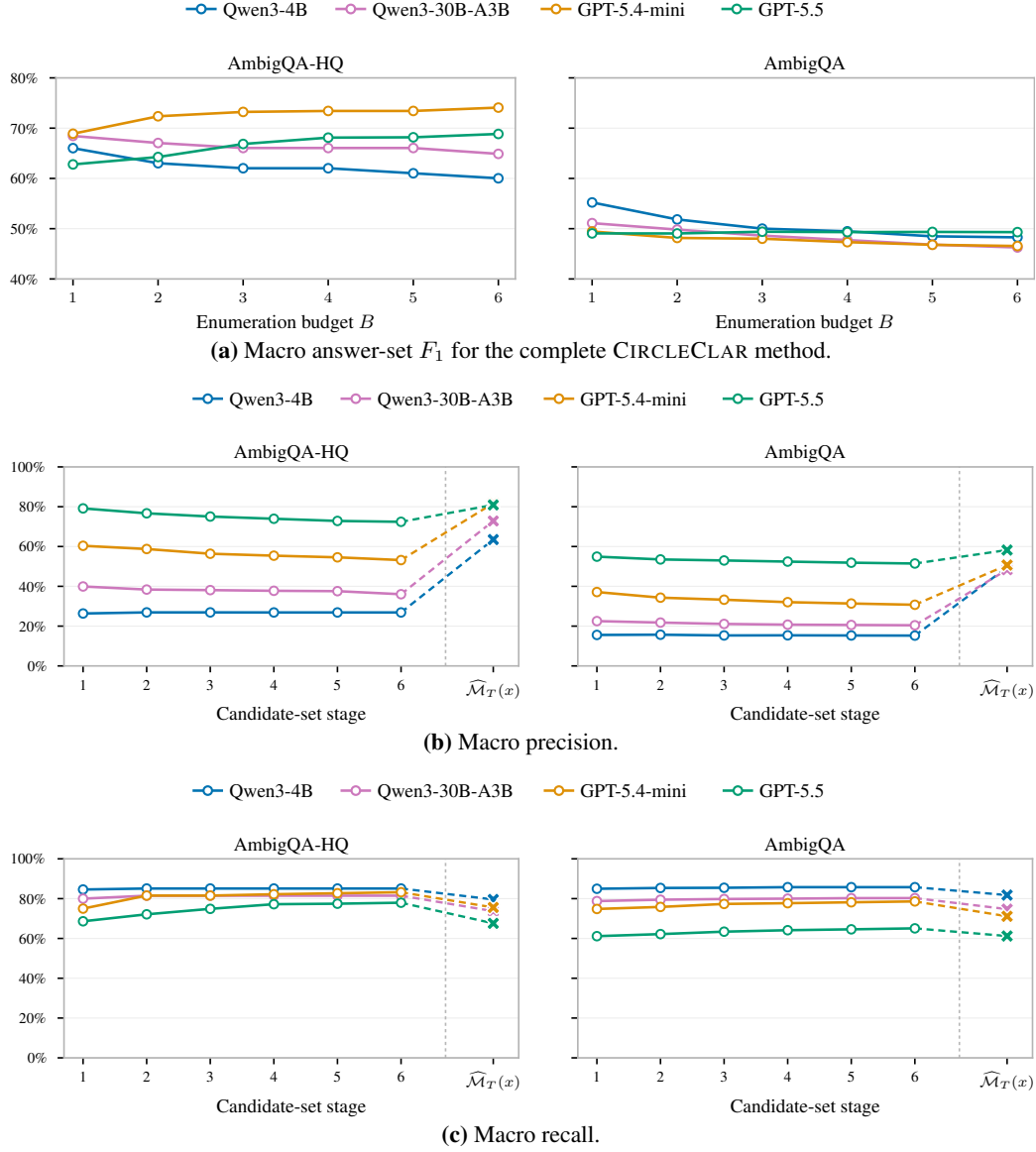


Figure 5: Effect of the enumeration budget and consistency check. Panel (a) varies B and reruns the complete CIRCLECLAR method at each value. Panels (b) and (c) instead report macro precision and recall along the candidate-accumulation trajectory for $B = 6$: the first five points correspond to $k = 1, \dots, 5$, $\mathcal{Y}_T(x)$ is the initial candidate set at $k = 6$, and $\widehat{\mathcal{M}}_T(x)$ is the final estimate after the consistency cycles.

D Content-Aware Evaluation of Set-Induced Response Policies

For an input x , let $\mathcal{M} = \mathcal{M}_T(x)$ be the model-realizable reference answer support and let $\widehat{\mathcal{M}} = \widehat{\mathcal{M}}_T(x)$ be a method’s estimated answer support. The reference support induces ABSTAIN, ANSWER, or CLARIFY when $|\mathcal{M}|$ is zero, one, or at least two, respectively. For any answer support S , let $a(S)$ denote the induced response action: ABSTAIN if $|S| = 0$, ANSWER if $|S| = 1$, and CLARIFY if $|S| \geq 2$.

Criteria. The main policy experiment in Section 5.2 evaluates whether the estimated and reference supports induce the same response action:

$$S_{\text{action}}(\widehat{\mathcal{M}}, \mathcal{M}) = \mathbb{1}[a(\widehat{\mathcal{M}}) = a(\mathcal{M})].$$

This action-only criterion permits a common comparison with Direct (Generic) and Direct (Model-Dependent), which predict an action without constructing an answer set. Here, we additionally evaluate whether methods that estimate an answer set recover content consistent with the reference support. We include the action-only criterion as a reference point and progressively strengthen the answer-content requirement. We rescore the same fixed estimates used in Section 5.1. Direct policies are omitted because they do not output answer sets.

The first content-aware criterion requires the correct unique answer in the ANSWER regime and at least two matched reference answers in the CLARIFY regime:

$$S_{\text{two}}(\widehat{\mathcal{M}}, \mathcal{M}) = \begin{cases} \mathbb{1}[\widehat{\mathcal{M}} = \emptyset], & |\mathcal{M}| = 0, \\ \mathbb{1}[\widehat{\mathcal{M}} = \mathcal{M}], & |\mathcal{M}| = 1, \\ \mathbb{1}[|\widehat{\mathcal{M}} \cap \mathcal{M}| \geq 2], & |\mathcal{M}| \geq 2. \end{cases}$$

For CLARIFY, this criterion permits additional predictions outside the reference support. A stricter supported-subset criterion additionally requires every predicted answer to be reference-supported:

$$S_{\text{subset}}(\widehat{\mathcal{M}}, \mathcal{M}) = \begin{cases} \mathbb{1}[\widehat{\mathcal{M}} = \emptyset], & |\mathcal{M}| = 0, \\ \mathbb{1}[\widehat{\mathcal{M}} = \mathcal{M}], & |\mathcal{M}| = 1, \\ \mathbb{1}[|\widehat{\mathcal{M}}| \geq 2 \wedge \widehat{\mathcal{M}} \subseteq \mathcal{M}], & |\mathcal{M}| \geq 2. \end{cases}$$

Finally, exact-set success requires complete recovery of the reference support:

$$S_{\text{exact}}(\widehat{\mathcal{M}}, \mathcal{M}) = \mathbb{1}[\widehat{\mathcal{M}} = \mathcal{M}].$$

The four criteria form the following ordering in strictness:

$$S_{\text{exact}} = 1 \implies S_{\text{subset}} = 1 \implies S_{\text{two}} = 1 \implies S_{\text{action}} = 1.$$

For each criterion, we compute the success rate separately within the three reference-cardinality regimes and report their unweighted mean, mirroring the balanced treatment of the three response actions in the main experiment.

Figure 6 shows that action-only rankings are mixed. Once answer identity is taken into account, CIRCLECLAR is generally strongest on AmbigQA-HQ, whereas results on the original AmbigQA are more mixed. The difference between the action-only and content-aware evaluations shows that predicting the correct response mode does not necessarily imply recovering the correct answers: a method may estimate the appropriate support cardinality while assigning that cardinality to unsupported content.

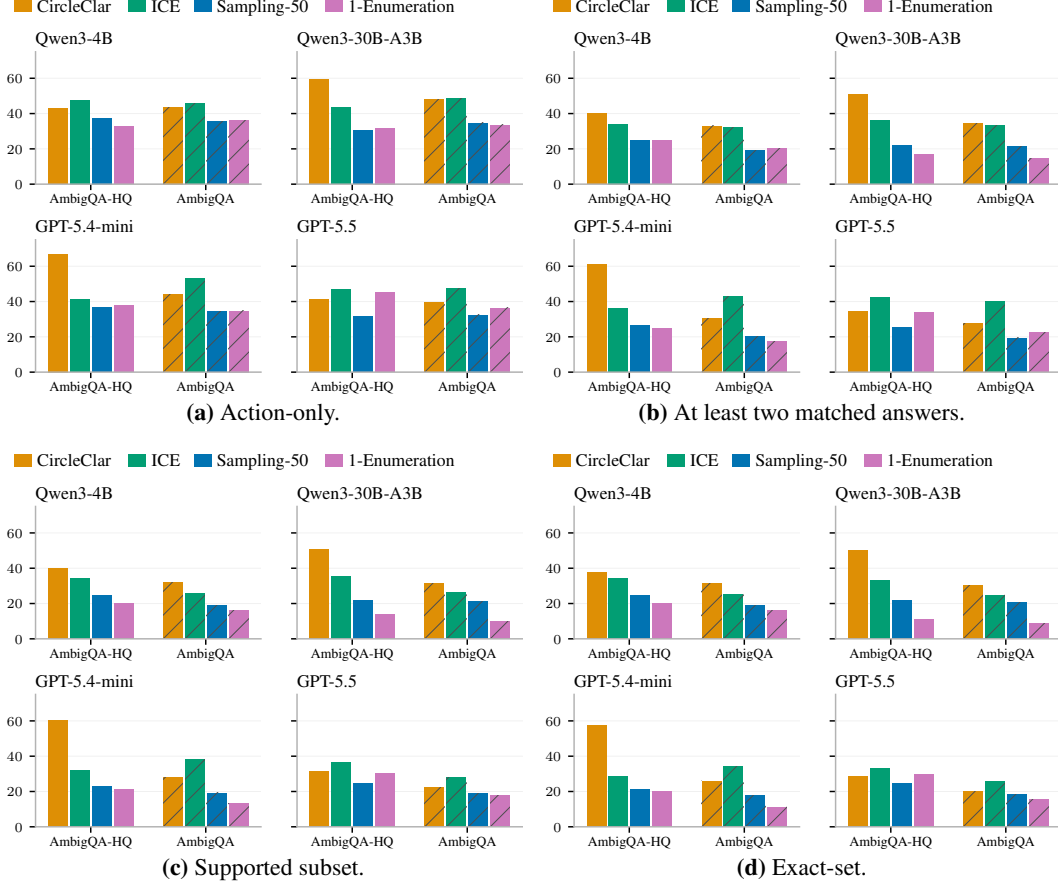


Figure 6: Content-aware response-policy evaluation of the four methods. Each panel reports the unweighted mean of the success rates in the ABSTAIN, ANSWER, and CLARIFY reference-cardinality regimes. Higher is better.

E Experimental Settings and Prompts

Overview. Across QA and text-to-SQL, we collapse identical outputs and use GPT-5.4-mini at temperature 0 for semantic grouping and equivalence judgment. Before scoring, we semantically group each method’s predictions under the original input and match them with the model-specific reference answer groups. We compute precision, recall, set F_1 , and exact set match (EM) from these matches. Within each task, the same equivalence judge is used during reference construction and the CIRCLECLAR consistency check. Task-specific generation and set-construction rules and prompts are given below.

E.1 Open-Domain Question Answering

Let m be the mass of the top re-answer group. Table 6 summarizes the QA generation budgets and set-construction rules.

E.1.1 Prompt Templates

ICE uses the released AmbigQA few-shot prompts of Hou et al. [5] without modification, so we do not duplicate them here.

Table 6: QA generation budgets and set-construction rules.

Method	Generation	Set construction
Reference construction	10 answers per dataset-provided rewrite, $T = 0.3$.	Accept if $m \geq 0.9$, and the top answer matches the candidate.
CIRCLECLAR	6 answer-set enumerations, $T = 0.5$; one clarification per candidate, $T = 0$; 10 re-answers per clarification, $T = 0.3$.	Apply the same consistency check as in reference construction to each candidate, with $m \geq 0.6$.
ICE	2 clarification samples, $T = 1.0$, and 8 answers per clarification, $T = 0.5$, using the released AmbigQA prompts.	Threshold answer counts separately within each clarification, union the groups, and sweep $t = 1, \dots, 8$.
Sampling-50	50 single-answer samples from the unresolved question, $T = 0.5$.	Discard UNKNOWN outputs, use the remaining sampled answers directly.
1-Enumeration	One answer-set enumeration, $T = 0.5$, using the CIRCLECLAR prompt.	Use the enumerated answers directly.

Answer-set enumeration (CIRCLECLAR and 1-Enumeration)

SYSTEM:

You answer open-domain questions by listing candidate final answers.

List all distinct final answers that could be correct under the original question or different reasonable readings of the original question.

Each item must be a standalone answer span, not a sentence or explanation.

Use the full canonical surface form for each answer.

For people, output full names, not initials, surnames, or shortened names.

For organizations, works, places, and events, output the full official or commonly used name, not an acronym or shorthand.

For full dates, use full month names. For counts or measurements, prefer digits.

Do not ask clarifying questions. Do not explain. Do not output aliases or formatting variants as separate answers.

Output exactly a numbered list:

1. answer
2. answer

USER:

Question: {question}

Answers:

Semantic grouping of predicted answers

SYSTEM:

You are a semantic clustering engine.

USER:

Task:

Cluster the answers by their semantic meaning for the given question.

First, silently identify a few semantic tags for each answer.

Then cluster answers that have the same core meaning.

Rules:

- Do not judge factual correctness.
- Do not add missing answers.
- Each answer must appear in exactly one cluster.
- Use as many or as few clusters as needed.
- Answers should be in the same cluster only if they would be treated as the same answer to the question.
- If two answers differ in date, number, or specificity, keep them separate.

Return JSON ONLY.

```
{
  "clusters": [
    {
```

```

        "cluster_id": 0,
        "member_indices": [0, 2],
        "cluster_tags": ["short tag"]
    }
]
}

Question:
{question}

Answers:
{numbered_answers}

```

Answer-conditioned clarification

SYSTEM:
You rewrite ambiguous open-domain questions into unambiguous questions.

Given an original question and a target answer, write one clarified question whose unique answer is the target answer under some reasonable interpretation of the original question.

Do not fact-check beyond choosing a reasonable interpretation. Do not explain. Do not include the target answer itself in the clarified question unless it already appears in the original question. Do not change the kind of answer the original question asks for.

The clarified question must not be identical to the original question.
The clarified question should add enough constraint that the target answer, rather than a generic or more common answer, would be the unique answer.

Output exactly one line:
Clarification: clarified question?

USER:
Original question: {question}
Target answer: {source_answer}

Clarification:

Resolved-question answering (reference construction and CIRCLECLAR)

SYSTEM:
You answer fact-seeking questions directly.

Return exactly one standalone final answer span, not a sentence, list, explanation, citation, or hedge.

Use the full canonical surface form for the answer.
For people, output full names, not initials, surnames, or shortened names.
For organizations, works, places, and events, output the full official or commonly used name, not an acronym or shorthand.
For full dates, use full month names. For counts or measurements, prefer digits.

If several aliases would refer to the same answer, choose one canonical form.
Do not output aliases or formatting variants as separate answers.

USER:
Question: {resolved_question}

Answer:

Sampling-50 direct answering

SYSTEM:
You answer open-domain factual questions directly.

Return exactly one short factual answer to the user's original question.
Do not list alternatives, ask a clarification question, explain your reasoning, or add a label.
Prefer a short canonical answer string such as a person, date, place, organization, title, number, or noun phrase.
If you cannot provide a plausible factual answer from your own knowledge, return exactly UNKNOWN.

Output exactly one non-empty line.

```

USER:
Question: {question}

Answer:

```

Prediction-group to reference-group judgment

```

SYSTEM:
You are an answer-equivalence judge. Do not answer the question yourself.

USER:
Question:
{question}

Canonical answer groups:
{canonical_groups_json}

Predicted semantic cluster:
{target_prediction_json}

The strings in the predicted cluster have already been grouped as one answer.
Decide whether that answer is equivalent to exactly one canonical group.
Treat the canonical groups as authoritative. Return NONE if it matches no group,
could match more than one group, contradicts a group, or is unclear.

Return JSON only, with 'group_id' set to one supplied group ID or 'NONE':
{"group_id": "G1", "reason": "one concise sentence"}

```

Pairwise answer-equivalence judgment (reference construction and CIRCLECLAR consistency check)

```

SYSTEM:
You are an expert semantic equivalence judge. Do not answer the question.

USER:
Question: {question}
Answer A: {ref_text}
Answer B: {candidate}

Do Answer A and Answer B convey the same essential answer to the question?

Return true only if they express the same core fact, entity, or value, regardless of phrasing or
formatting.
Aliases, former names, stage names, pen names, nicknames, and character identities count as equivalent
when they identify the same entity.
Return false if they differ, contradict each other, are merely related, or equivalence is unclear.

Respond as JSON:
{"score": true or false, "reason": "one sentence"}

```

E.2 Executable Text-to-SQL

Let m be the mass of the top re-answer group. Table 7 summarizes the SQL generation budgets and set-construction rules.

Table 7: Text-to-SQL generation budgets and set-construction rules.

Method	Generation	Set construction
Reference construction	5 SQL samples per dataset-provided resolved interpretation, $T = 0.2$.	Accept if $m \geq 0.8$ and the top answer matches the candidate.
CIRCLECLAR	5 answer-set enumerations, $T = 1.0$; one clarification per grouped candidate, $T = 0$; 5 recovery queries per clarification, $T = 0.2$.	Execute and semantically group the candidates; apply the same consistency check with $m \geq 0.6$.
ICE	2 clarification samples, $T = 1.0$, using up to 8 distinct rewrites (or the original request if none parse); 8 SQL samples per rewrite, $T = 0.5$.	Threshold answer counts separately within each clarification, union the groups, and sweep $t = 1, \dots, 8$.
Sampling-50	50 single-query samples from the unresolved request, $T = 0.5$.	Use the sampled answers directly.
1-Enumeration	One answer-set enumeration, $T = 1.0$, using the CIRCLECLAR prompt.	Use the enumerated answers directly.

E.2.1 Prompt Templates

SQL answer-set enumeration (CIRCLECLAR and 1-Enumeration)

The request might be ambiguous. These ambiguities can arise from various factors, including but not limited to:

1. Ambiguous references to entities in the request.
2. Multiple properties of objects or entities leading to different interpretations.
3. Unclear timestamps or time conventions.
4. Unclear locations or spatial scope.
5. Multiple valid answer types based on the request.

The task is to write SQLite queries based on the provided request. The request can be ambiguous and admit different reasonable interpretations. Write all SQL queries corresponding to the reasonable interpretations you can identify, and separate each SQL query with an empty line.

Do not include explanations, labels, Markdown fences, or duplicate/formatting variants. Do not select extra columns beyond those requested. Each query must be a single read-only SELECT or WITH statement.

Given the following SQLite database schema and content:

DATABASE_DUMP

Answer the following unresolved request:

UNRESOLVED_REQUEST

Return only the SQL queries, separated by empty lines.

Semantic grouping of SQL answers

Partition the SQL outputs by the reasonable interpretation of the input request that they answer.

Compare meaning, not table shape. Merge outputs with the same requested entities, filters, scope, grouping or aggregation, time convention, alternatives, and facts even when one adds ancillary columns, reorders columns, changes SQL form, or uses a lossless vertical/horizontal layout.

Critical contrast: if an ambiguous phrase has plausible facts A and B, keep A-only, B-only, and A+B as separate interpretations. An added field is ancillary only when it is context; it is not ancillary when it realizes another live reading. Sharing a column or having a lossless projection is therefore evidence, not the decision.

Use the input, schema, SQL, and complete typed results together. Assign each output to exactly one cluster. If semantic equivalence remains uncertain, keep the outputs separate.

Return only JSON:

```
{
  "clusters": [
    {
      "members": ["id1", "id2"],
      "interpretation": "short natural-language description"
    }
  ]
}
```

Input request:

```
{{INPUT_REQUEST}}

Database schema:
{{DATABASE_SCHEMA}}

SQL and results:
{{SQL_RESULTS}}
```

Answer-conditioned SQL clarification (CIRCLECLAR)

SYSTEM:
You rewrite ambiguous database requests into unambiguous requests.

Given an original request and a target query, write one clarification that expresses the query's interpretation as a reasonable reading and rules out the other reasonable readings.

Do not explain or mention SQL, schema names, or result values. Do not change what kind of output the original request asks for.

The clarification must not be identical to the original or merely cosmetic.
Never output REJECT.

Output exactly one line:
Clarification: clarified request

USER:
SQLite database context:

DATABASE_DUMP

Original request:
UNRESOLVED_REQUEST

Target query:
REPRESENTATIVE_SQL

Clarification:

Resolved-request SQL generation (reference construction and CIRCLECLAR)

The task is to write exactly one SQLite query for the resolved request below. The request has already been disambiguated. Do not include explanations, Markdown fences, or alternative queries, and do not select extra columns beyond those requested.

Given the following SQLite database schema and content:

DATABASE_DUMP

Resolved request:
RESOLVED_REQUEST

Return exactly one SQL query.

Direct SQL generation (Sampling-50 and ICE answering)

You are given a SQLite database dump and one database request.

Database dump:
DATABASE_DUMP

Request:
REQUEST

Write exactly one read-only SQLite SELECT or WITH query that answers the request.
Return only the SQL query, with no prose, label, or Markdown fence.

ICE clarification generation

The request might be ambiguous. These ambiguities can arise from various factors, including but not limited to:

1. Ambiguous references to entities in the request.
2. Multiple properties of objects or entities leading to different interpretations.
3. Unclear timestamps or time conventions.
4. Unclear locations or spatial scope.
5. Multiple valid answer types based on the request.

You are given one unresolved natural-language database request and the complete SQLite database dump available to the answering model.

Database dump:
DATABASE_DUMP

Unresolved request:
UNRESOLVED_REQUEST

Provide one distinct rephrasing for each reasonable reading of the unresolved request that you can identify. By "rephrasing," we mean reformulating the request to be clear and direct, eliminating ambiguity without altering its original intent. Each rephrasing must select exactly one reasonable reading, preserve the requested output, and be answerable from this database.

Do not seek further information, write SQL, answer the request, mention ambiguity labels, or add a reading unsupported by the wording and database. Return at most eight rephrasings. If the original request is already clear and unambiguous, return an empty list.

Return exactly one JSON object with this schema and no other text:
{"clarifications":["first rewritten request","second rewritten request"]}

SQL prediction-group to reference-group judgment

Judge whether one predicted SQL answer group is equivalent to exactly one reference answer group. Do not answer the request or write SQL.

The predicted group has already been semantically grouped and should be judged as one answer. Use the request, schema, SQL, and complete typed results. Compare meaning, not SQL syntax or table shape. Allow ancillary columns, column reordering, and lossless reshaping only when the requested entities, filters, scope, grouping or aggregation, time convention, alternatives, and facts are preserved. Do not accept accidental result equality when the SQL meanings differ.

Treat the reference groups as authoritative. Return NONE if the prediction matches no group, could match more than one group, changes a requested fact or interpretation, or is unclear.

Return only JSON, with group_id set to one supplied group ID or NONE:
{"group_id":"G1","reason":"one concise sentence"}

Input request:
{{INPUT_REQUEST}}

Database schema:
{{DATABASE_SCHEMA}}

Reference answer groups:
{{REFERENCE_GROUPS_JSON}}

Predicted answer group:
{{PREDICTED_GROUP_JSON}}

Consistency-check judgment (reference construction and CIRCLECLAR)

Judge whether re-answering the resolved request preserves the source answer on this fixed database.

The source SQL and result are authoritative; never correct or replace them. Use the resolved request, schema, SQL, complete typed results, and deterministic relation checks. Treat the relation checks as evidence, not as the final semantic decision.

Return equivalent only when every requested source fact is preserved. Allow ancillary-column differences and lossless reshaping or derivation. Return not_equivalent if any requested source

fact, row, entity, filter, scope, grouping or aggregation, time convention, or live alternative is missing, replaced, or changed. Do not accept accidental result equality when the SQL meanings differ. Return uncertain only when the evidence is insufficient.

Return only JSON:

```
{"decision": "equivalent|not_equivalent|uncertain", "reason": "short reason"}
```

Resolved input request:

```
{{RESOLVED_REQUEST}}
```

Database schema:

```
{{DATABASE_SCHEMA}}
```

Deterministic relation checks:

```
{{DETERMINISTIC_RELATION_CHECK}}
```

Source SQL and result:

```
{{SOURCE_SQL_RESULT}}
```

Stable top-mass recovery SQL and results:

```
{{STABLE_RECOVERY_SQL_RESULTS}}
```

E.3 Direct Response-Policy Baselines

Both direct baselines receive only the unresolved question, select a response action without constructing an answer set and are queried once per input at temperature $T = 0$. We parse the predicted action from the action field.

Direct (Generic) response-policy baseline

SYSTEM:

You are an AI assistant choosing how to respond to a user's factual question. Do not answer the question itself and do not write a clarification question. Return only the requested JSON object, with no Markdown or extra text.

USER:

Choose exactly one conversational action for the user's question: ANSWER, CLARIFY, or ABSTAIN.

Use these conversational criteria:

- ANSWER when the question is sufficiently clear and you know the answer.
- CLARIFY when ambiguity or missing details prevent a direct answer, and the user can resolve the issue by specifying their intent without needing to know the factual answer themselves.
- ABSTAIN when the question is sufficiently clear but you do not know the answer.

Choose the action you would take now from the question alone. Do not assume access to search, tools, reference answers, or additional evidence.

Return exactly this JSON schema:

```
{"action": "ANSWER|CLARIFY|ABSTAIN"}
```

User question: {question}

Direct (Model-Dependent) response-policy baseline

SYSTEM:

You are an AI assistant deciding how to respond to a user's factual question.

Do not answer the question itself.

Do not write a clarification question.

Return only the requested JSON object, with no Markdown or additional text.

USER:

Choose exactly one action for the user's question:

ANSWER, CLARIFY, or ABSTAIN.

Base your decision only on the question and your own current knowledge.

Use the following criteria:

- ANSWER if you can reliably provide exactly one distinct answer. Choose ANSWER even if the question has other plausible readings, as long as you cannot reliably provide a different answer for any of them.

- CLARIFY if at least two plausible readings of the question lead to different answers that you can reliably provide, so the user's intended reading is needed before choosing an answer.

- ABSTAIN if you cannot reliably provide an answer under any plausible reading of the question.

Do not choose CLARIFY merely because the wording allows multiple readings. The decision should depend on the different answers you can reliably provide, not only on ambiguity in the wording.

Do not assume access to search, tools, reference answers, or additional evidence.

Return exactly:
{"action": "ANSWER|CLARIFY|ABSTAIN"}

User question: {question}