THE ATTRIBUTE TRAP: SYSTEMATICALLY SUBOPTIMAL CLARIFYING QUESTIONS AFTER TOOL-CALL RETRIEVAL

Anonymous authors
Paper under double-blind review

ABSTRACT

Do task-oriented dialogue agents ground their clarifying questions in the records they have just retrieved? We study *closed-set candidate disambiguation*: a task-oriented dialogue agent retrieves multiple candidate records (orders, bookings, search results returned by a database query or an application programming interface (API) call) matching a user's request, and must ask a single clarifying question to identify which one of the N candidates is meant. Because the candidate set is known, the expected information gain of every attribute the agent could ask about can be computed exactly, so an agent's question can be scored by its regret against the most informative one. This lets us separate genuine judgment failures from three confounds that distort naive failure rates: candidate sets that no natural attribute can resolve, questions that were in expectation as good a bet as the best one, and near-identifier attributes that are informative whatever the table contains. Our main evidence is interventional. On real candidate sets from the Schema-Guided Dialogue corpus (SGD) and the Olist e-commerce data we edit a single column and leave every other cell unchanged. The edits split nine open-weight instruction-tuned models (7–32B parameters) in two. When the column a model asks about most is only partly duplicated, so that it is still informative but no longer the best, four models, among them the two largest, stop asking it (3–21% of draws); four keep asking on 73–85%; one lies between. When the column is flattened so that it carries no information, six stop, two fall to about the chance rate, and one keeps asking at twice it. So four models do not react to the values in the table; four do, in different degrees. We call asking about an attribute that is not the most informative one an **attribute trap**. On unedited tables from four dataset families (SGD, the Multi-Domain Wizard-of-Oz corpus (MultiWOZ), the τ-bench family, Olist) every model falls into it; for the strongest models the loss is small and sits almost entirely on attributes such as an exact price, which are the most informative but which a real user may not know. The effect persists across prompt wordings, a recency-weighted prior and scoring thresholds, and it is not explained by counting errors alone: giving the agent the exact number of distinct values of every attribute does not reliably repair it, and stating the goal makes most models worse.

1 INTRODUCTION

A customer asks a booking agent about "my movie ticket". The agent's tool call returns four showings that differ in price ($23, $11, $16, $21) and in format (three *regular*, one *IMAX*). Asking for the price identifies the ticket in one turn; asking "was it an IMAX or a regular showing?" leaves three candidates whenever the answer is *regular*—which it is here. On this real candidate set from the Schema-Guided Dialogue data (Rastogi et al., 2020), four of the nine instruction-tuned models we study (Qwen2.5, Llama 3, Gemma 2, GLM-4, Mistral, Yi, DeepSeek; Qwen Team, 2025; Llama Team, AI @ Meta, 2024; Gemma Team, 2024; GLM Team, 2024; Jiang et al., 2023; 01.AI, 2024; DeepSeek-AI, 2024), among them a 14B model (B: billions of parameters) from the top tier, ask about the format in five out of five samples (Fig. 1). We call this an **attribute trap**.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

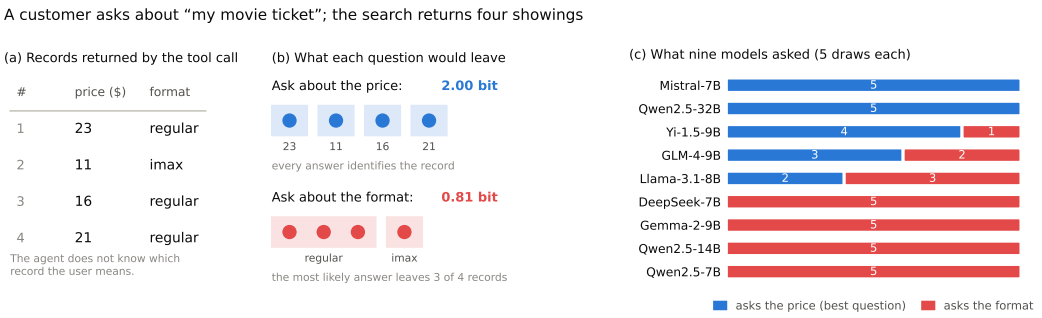


Figure 1: One benchmark task, end to end (SGD Movies, real retrieval result). (a) The four records the tool call returned. (b) Under a uniform prior, asking about the price separates all four records (2.00 bit), while asking the format most likely leaves three (0.81 bit); regret is the difference, 1.19 bit. (c) The attribute each of nine models chose to ask about in five samples: four models ask the format every time. The benchmark consists of 238 such tables.

This situation—*closed-set candidate disambiguation*—is routine in task-oriented dialogue: in about two thirds of MultiWOZ (Budzianowski et al., 2018) and SGD dialogues a database search returns several matching entities, so an opportunity for disambiguation arises (Qian et al., 2022). It is also measurable. Because the candidate records are on the table, the expected information gain (Lindley, 1956) of every question the agent could ask is computable exactly, with no learned belief model and no human annotation. We can therefore ask a sharper question than “did the dialogue succeed”: *is the agent’s clarifying question grounded in the records it has just retrieved?*

Answering it requires care, because naive failure rates are distorted in three ways. Some candidate sets cannot be resolved by any natural attribute. Some questions are equally informative in expectation, and whether they resolve the target is chance. And some attributes, a movie’s title, one flight per airline, are distinct for every candidate by construction of the data, so asking them does not establish adaptation to this table’s particular values; these turn out to be what models ask most. We define a regret-based score and one uniform rule for each confound (§3), and show that the conclusions do not depend on the thresholds involved.

On unedited tables every one of nine open-weight models (7–32B) asks non-optimal questions, the worst with 9.0–11.3$\times$ the regret of the best (§4.1), also on a second sample of unmodified tables fixed before any model was run on it (0.11–0.58 bit). But a rate of non-optimal questions cannot say whether a model read the table and preferred another attribute, or never read it. Our main evidence is therefore interventional: paired edits of a single column in real candidate tables (§4.3).

Benchmark and findings. To measure this we build AttrTrap: 238 real candidate tables from four public corpora, each scored against the exact information-gain optimum with information-tied and near-identifier cases set aside, paired single-column and column-name edits of the same tables, and a second sample of 251 tables whose selection rules were fixed before any model was run. On it we find: **(1) Weak models do not read the table; strong ones do.** Editing one column of a real table splits nine open-weight models: four keep asking about a column after it has stopped separating the candidates, one even when it separates nothing, while four, among them the two largest, switch (§4.3). **(2) Knowing is not doing.** Handing the model the exact number of distinct values of every attribute, telling it to count them, or stating the goal repairs no model reliably: the counts, the instruction to count and the stated goal make 2, 4 and 7 of 9 models significantly worse (§4.4): the failure is not a missing fact or a misunderstood objective. **(3) Information-optimal is not the same as askable.** The regret that remains for the strongest models sits almost entirely on attributes such as an exact price, which separate the candidates best but which a caller may not know; stating the goal does not move them (§5). We release AttrTrap, the raw outputs of every model call and a deterministic scoring pipeline.

2 Related work

Choosing questions by expected information. Selecting the question with the highest expected information is a classical idea (Lindley, 1956): it is the split rule of decision trees, whose bias toward many-valued attributes (Quinlan, 1986) is the near-identifier problem of §3.1, and it chose database questions in dialogue systems before neural models (Li et al., 2016). Rao & Daumé III (2018) brought it to clarification questions with a neural expected-value-of-perfect-information (EVPI) ranker. With large language models (LLMs), questions are chosen by simulated expected information gain (EIG) (Hu et al., 2024), by Bayesian experimental design (Kobalczyk et al., 2025; Choudhury et al., 2026), amortised with reinforcement learning (Hartmann et al., 2026), over tool-call parameters (Suri et al., 2026), or trained with an information-gain reward on a clarification version of τ-bench (Deng et al., 2026a); Mazzaccara et al. (2024) fine-tune on EIG-ranked question pairs, and Harne et al. (2026) compute the entropy of each attribute over the current candidates in product search as one component of their method. These *methods* replace the model's free choice with a computed one; none measures the free choice of an off-the-shelf agent on retrieved records, and our oracle-count result is a caution for them: supplying the right statistic is not the same as the model using it.

Do language models ask informative questions? In 20 questions, ChatGPT approaches the optimal strategy only when made to list the remaining hypotheses (Bertolazzi et al., 2023); in a collaborative Battleship game language-model (LM) agents ask less informative questions than people (Grand et al., 2026); reinforcement learning for active reasoning can lock agents into uninformative querying (Zou et al., 2026); and models must be trained to ask at all when a request is ambiguous (Zhang et al., 2025). These works use games or synthetic hypothesis spaces with a learned or sampled belief; we study the first clarifying question over a real retrieval result, where the information content of every candidate question is known exactly, so judgment can be separated from chance and the tables themselves can be edited. Clarification benchmarks chiefly evaluate *whether* an input is ambiguous and *whether* and *when* to ask (e.g. Gulati et al., 2026; Zhao et al., 2026; Tao et al., 2026; Baan et al., 2026; Deng et al., 2026b); we evaluate *what* is asked.

Reading tables. How well LLMs read a small table depends on its rendering (Sui et al., 2024), and models that parse the structure still mis-cite values (Yang et al., 2026); our markdown paraphrase and our edits ask the same of a table the agent must act on.

Clarification in search and recommendation. Asking a clarifying question before retrieving is a standard move in conversational search (Aliannejadi et al., 2019; Zamani et al., 2020), and conversational recommenders choose which attribute to ask about to narrow a candidate set (Christakopoulou et al., 2016; Lei et al., 2020); there a trained policy chooses the attribute, here we ask whether a language model does so unaided. Pragmatic models of reference make the trade-off between the informativeness of an utterance and its cost explicit (Frank & Goodman, 2012); that a question should also be *answerable* by this user is our extension of that insight, and regret leaves it out (§5).

Disambiguating database results in task-oriented dialogue. Qian et al. (2022) study database-search-result disambiguation, show that the opportunity for it arises in roughly two thirds of MultiWOZ (Budzianowski et al., 2018) and SGD (Rastogi et al., 2020) dialogues—the corpora themselves usually avoid the exchange by listing one matching result—and train dialogue models to resolve the user's choice when the system *lists* the matching options. We study the complementary case in which the agent instead asks about a natural attribute, as agents built on tool-use benchmarks such as τ-bench (Yao et al., 2024; Barres et al., 2025) do. Our candidate sets are taken from the databases of these corpora and from the Olist e-commerce data (Olist, 2018).

3 Setting and measurement

An agent has retrieved $N \in [2, 8]$ candidate records via a tool call. Each record exposes natural attributes $\mathcal{A}$; the agent must choose one $a \in \mathcal{A}$ to ask about (system IDs are disallowed). The model sees only the records and returns a question plus the attribute it asks about, which we score (prompts: Appendix B); on the runs whose full logs we kept, the declared attribute is what the question asks

about (Appendix A). With a uniform prior over the N candidates, and assuming the user answers truthfully so that the answer reveals the target's value of a, the expected information gain (EIG) of a is the entropy of the partition it induces:

$$p_a(v) = \frac{|\{i : x_{i,a} = v\}|}{N}, \qquad \mathrm{EIG}(a) = -\sum_v p_a(v) \log_2 p_a(v), \qquad R(a) = \max_{a' \in \mathcal{A}} \mathrm{EIG}(a') - \mathrm{EIG}(a), \tag{1}$$

where $x_{i,a}$ is the value of attribute a in candidate i. We score a choice by its **regret** $R(a) = \mathrm{EIG}^* - \mathrm{EIG}(a)$ in bits, where $\mathrm{EIG}^* = \max_{a'} \mathrm{EIG}(a')$. We report the mean regret of a model and its *regret-0* share, the fraction of draws with zero regret. Each task is sampled k=5 times at decoding temperature T=0.3 (five *draws*); all 95% confidence intervals (CIs) are task-level bootstrap intervals (2000 resamples), because draws within a task are not independent.

Why not "did the question resolve it". The agent never sees which candidate is the target, so two attributes with identical partitions are equally good bets and landing on the singleton side is chance. On SGD, 37 of 129 (28.7%) outcome-level failures of Qwen2.5-14B are exact EIG ties. We therefore exclude *information-tied* tasks, defined in §3.1, from every metric.

3.1 Benchmark construction

Every candidate set in the benchmark is taken from existing data; no record is written by us. Table 13 counts each step and Table 14 gives it per source.

Step 1: candidate sets (4,636). *SGD* (train split, files 001–020; 2,559 single-domain dialogues, 9 services): every system turn whose database call returns 2–8 results; the returned records are the candidates, and the attributes are the result fields minus the user's own search constraints (identical across candidates by construction) and name, address and phone fields. *Olist*: the orders of one real customer, for the 2,997 customers with at least two orders (2,797 pass Step 2). τ*-bench* retail and airline: the records of a user whose task requires identifying one of them (43 and 7 tasks); the τ^3-bench banking tables are treated the same way (21 users). *MultiWOZ*: venues matching a two-constraint search (area and food, or area and type) with 2–8 matches (35 groups, 34 after Step 2). The target is the record designated by the task in τ-bench and the first record elsewhere; regret does not depend on the target, but the two filters below do: 217 of the 238 benchmark tables pass them whichever candidate is the target, and restricting to these moves no model's mean regret by more than 0.011 bit and changes no rank (Table 24).

Step 2: pool (4,133). We keep candidate sets in which at least one attribute isolates the target; in the other 503 no question about a natural attribute can succeed.

Step 3: source-capped sample (292). Pool sizes range from 7 to 2,797, so we draw uniformly at random within each source up to a cap (15 by default; 60 for Olist, 20 for SGD Movies, 10–11 for the MultiWOZ and banking pools; τ-bench taken whole; Table 14). The caps limit source imbalance rather than estimate a deployment distribution; re-weighting sources changes no conclusion (§3.2).

Step 4: near-identifier attributes (-106, $+76$). Some attributes are distinct for every candidate by construction of the data (one flight per airline; a movie's title), so asking them is optimal whatever the table says and demonstrates no table reading. One rule is applied to all 17 sources: an attribute is near-ID if all candidates differ on it in $\geq 90\%$ of the source's $N \geq 3$ sets. It flags attributes in six sources (Table 5), and these are what models ask most (`title` 89% of all picks, `card_type` 91%, `airlines` 75% in SGD Flights_1, `item_names` 73%). MultiWOZ `postcode` (83%) and SGD Events `date` (84%) fall below the line and are kept. We formulated this rule after observing that models overwhelmingly asked such attributes; it is computed from the tables alone, and we also report results without it (all 17 sources, Table 7; other thresholds, Table 6). The 106 sampled tasks of the flagged sources are not scored. In their place we use 76 tasks rebuilt from the same sources' pools with the flagged columns removed and no cell edited (up to 25 per source, $N \geq 3$ first, kept only if an attribute still isolates the target); two sources have no usable task left and are dropped.

	mean regret (bit) [95% CI]	regret-0	non-opt. tables
Qwen2.5-32B	0.036 [0.019, 0.056]	93.8%	6.3%
Qwen2.5-14B	0.061 [0.035, 0.090]	91.2%	9.2%
Mistral-7B	0.076 [0.050, 0.105]	87.8%	12.2%
Qwen2.5-7B	0.079 [0.051, 0.109]	87.9%	13.9%
Llama-3.1-8B	0.088 [0.056, 0.123]	88.6%	13.9%
Gemma-2-9B	0.105 [0.069, 0.146]	86.1%	14.3%
GLM-4-9B	0.129 [0.097, 0.164]	84.1%	26.1%
Yi-1.5-9B	0.172 [0.130, 0.216]	78.0%	24.8%
DeepSeek-7B	0.406 [0.347, 0.469]	57.3%	49.2%
uniform random attribute	0.690	42.3%	–

Table 1: Final benchmark, 238 unique candidate tables, k=5 draws per task. Regret is the expected information lost relative to the best available question; intervals are task-level bootstrap. Regret-0: share of draws with zero regret; non-opt. tables: share of tables with at least one non-optimal draw. The last row is a uniform-random reference on the same tables.

Step 5: information-tied tasks (-7). A task is information-tied if the best attribute that isolates the target is less than 0.2 bit more informative than the best one that does not, so the table does not single out a best question. Such tasks are excluded from every metric.

Step 6: identical tables (-17). Different task ids can show the model exactly the same table (several τ-bench tasks of one user; the same search in different SGD dialogues). Regret does not depend on the target, so these are one question asked several times; we merge them into one unit and pool their draws.

The benchmark is the remaining 238 unique tables from 15 sources: 171 unmodified and 67 rebuilt. Because rebuilt tasks modify the data, results on the 171 unmodified tables alone are given in §4.1 and Tables 3, 2, 15; the prompt, repair and decoding experiments use only unmodified tasks. SGD contributes 125 of the 238 tables and most of the difficult ones; Olist, τ-bench and MultiWOZ serve as cross-domain checks (Olist's pool consists almost entirely of two-candidate sets; see the appendix). Each model is scored on the tables for which it produces a parsable answer (all 238, except Mistral-7B with 237), so no row depends on which other models are evaluated. The tasks are not selected from any model's failures.

A second sample. Because the benchmark is dominated by two-candidate tables and its harder part is rebuilt, we also draw a second sample that contains neither: sources without a near-ID attribute only (no table is modified), unique tables, $N \geq 3$, not information-tied, and at least one non-optimal attribute, so that a wrong choice is possible; at most 60 tables per source, uniformly at random (251 tables, 11 sources; Table 18). These six rules read the tables only and were committed to the repository before any model was run on the sample. It is drawn from the same pools as the benchmark, and 51 of its 251 tables also occur there; on the 200 that do not, mean regret is the same or higher for eight of the nine models.

3.2 Robustness of the construction

None of these choices drives the conclusions (details in Appendix A). Moving the tie threshold over $\{0, 0.1, 0.2, 0.3\}$ changes no model's mean regret by more than 0.006 bit and no rank (Table 12). Results without the near-ID rule are given in §4.1. Re-weighting sources moves no model's mean regret by more than 0.03 bit except DeepSeek-7B (0.406 $\rightarrow$ 0.322 and 0.301), whose errors are concentrated in the largest source.

4 RESULTS

4.1 REAL FOR EVERY MODEL, STRONGLY MODEL-DEPENDENT

Every model asks non-optimal questions (Table 1); the worst model has 9.0–11.3× the regret of the best, depending on how sources are weighted (§3.2). A paired permutation test on task-level differences, Holm-corrected over all 36 pairs, gives tiers rather than a strict order: DeepSeek-7B is worse than all eight others, Yi-1.5-9B worse than the top five, GLM-4-9B worse than the two largest Qwen models, Qwen2.5-32B better than every model except Qwen2.5-14B, and the other pairs among the six models in between indistinguishable (19 of 36 pairs significant, Table 9). GPT-4o (OpenAI, 2024), scored the same way (Table 17), would rank sixth (0.101 bit). The regret of the six stronger models lives almost entirely at $N \geq 3$ (Table 8): at $N{=}2$ (45% of tables) they are essentially perfect, whereas DeepSeek-7B is *worst* there (0.539 bit), asking attributes on which the two records do not differ. Where they do lose information is telling: on rebuilt SGD Flights_2 six models share a regret of 0.183 bit, all asking the departure time where the fare is optimal. The ordering is unchanged under a recency-weighted prior (the candidate with the largest value of the table's date-like attribute gets twice the weight of every other candidate, normalised; tables with no date-like attribute keep the uniform prior): mean regret moves ≤ 0.012 bit. Without the near-ID rule (247 original tasks, all 17 sources) every model still has non-zero regret (0.027–0.369 bit); DeepSeek-7B and Yi-1.5-9B remain last and Qwen2.5-32B first, but the order in between changes (Table 7), so we claim no order within the top tier.

Unmodified, non-trivial tables. Is the regret of the strongest models an artefact of the rebuilt tables? On the unmodified part of the benchmark (171 tables, 108 of them with two candidates) they are nearly perfect (Qwen2.5-32B 0.013, Qwen2.5-14B 0.022 bit). On the second sample, where every table is unmodified and a wrong choice is always possible, they are not: mean regret is 0.109–0.141 bit for the best three models and 0.577 for DeepSeek-7B, 14–57% of tables draw a non-optimal question (Table 18). The tiers are the same as on the benchmark and sharper (24 of 36 pairs significant after Holm correction; Table 20). Regret grows with the size of the candidate set (Qwen2.5-32B: 0.026, 0.115 and 0.204 bit at $N = 3$, 4 and ≥ 5). For the strongest models it is concentrated in SGD Movies (0.37–0.44 bit; Table 25), where the best attribute is the price on 50 of 57 tables; with all sources weighted equally their mean regret is 0.044–0.073 bit, still above zero (Table 19).

4.2 NOT A PROMPT ARTEFACT

We re-ran all models under three paraphrases (Table 3): terse with no persona; a different persona with the goal stated as "identify the entry with as few questions as possible"; and the original wording with records rendered as a markdown table. Of 27 model×paraphrase cells, 23 have *higher* regret than the original prompt and none of the four lower ones is significant: most wordings are worse (e.g. Qwen2.5-14B 0.022 → 0.105 bit under the second paraphrase). DeepSeek-7B is last under every wording; the order among the others changes with it. Magnitudes are prompt-dependent, the effect is not, and the swings are large: the "fewest questions" wording moves the 32B model from 0.013 to 0.149 bit. The additional errors under paraphrase fall on the salient attribute at about the chance rate, so they are not a salience effect. Nor is the effect a sampling artefact: under greedy decoding every model still has non-zero regret, and at $T{=}1.0$ no model improves significantly while three get worse (Table 15).

4.3 WHAT KIND OF FAILURE? PAIRED COUNTERFACTUALS

For each source, call the attribute asked most, pooled over all models, *salient*; it is fixed per source before any table is edited and is the same for every model. As in minimal-pair diagnostics (e.g. Ma & Miyao, 2026), we take tasks ($N \geq 3$, near-ID-free sources) in which the salient attribute is optimal and edit *one column*, leaving every other cell unchanged; the unedited task provides the paired "before" (Fig. 2, Table 4; a reverse edit is in Appendix A).

Extreme edit (65 tables): the column is flattened, so the attribute carries zero information and asking it is the worst available choice. Five models stop asking it entirely and Mistral-7B nearly

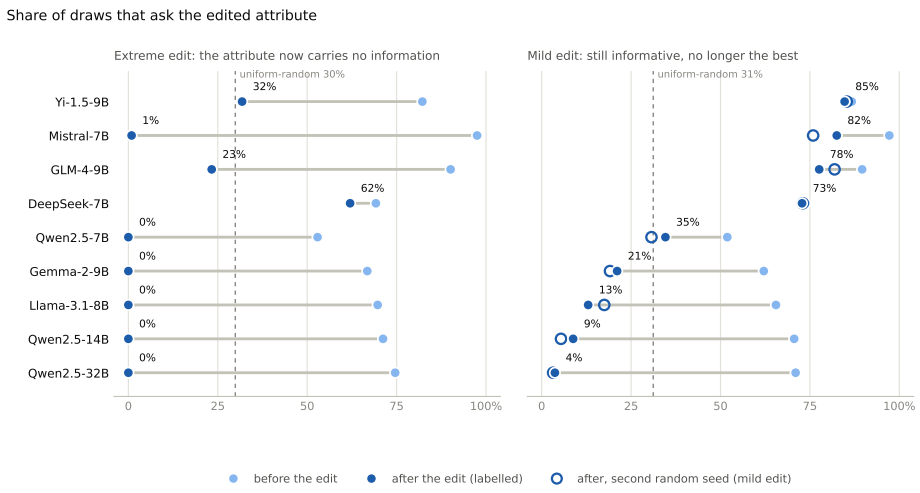


Figure 2: Does a model stop asking an attribute once it stops being useful? Each line runs from the share of draws asking the salient attribute before the edit (light) to after it (dark; open circle: second random seed). Left: after the column is flattened, five models stop completely; DeepSeek-7B barely moves. Right: after a milder edit, four models stay far above the uniform-random rate. Exact values and intervals are in Table 4.

so (0.9% of draws). DeepSeek-7B still asks it on 62.0% [50.3, 73.2] of draws against a 29.9% uniform-random baseline, and its regret-0 share falls from 75.1% to 7.7%. GLM-4-9B (23.3%) and Yi-1.5-9B (31.8%) are at about the chance rate yet lose half their accuracy: they abandon the attribute without finding the best remaining one.

Mild edit (57 tables, two seeds): about half of the other candidates receive the target's value, so the attribute stays somewhat informative but is no longer optimal. Models split: Qwen2.5-32B, Qwen2.5-14B, Llama-3.1-8B and Gemma-2-9B ask it on 3–21% of draws; Mistral-7B, DeepSeek-7B, GLM-4-9B and Yi-1.5-9B on 73–85% (regret-0 after the edit 1–24%); Qwen2.5-7B lies between (31–35%), as does GPT-4o (45–49%); the seeds agree within 7 points. As the edited attribute is often the runner-up, this shows a failure to notice that it stopped being the best, not by itself a table-independent default.

Names or values? (Table 22). Two edits keep every cell and change only the column names. *Swap*: the salient column and the least informative one exchange names, so the salient name heads values that barely separate the candidates. No model follows the values fully: Qwen2.5-14B asks the partner name on 39% of draws (regret-0 86% → 55%); Mistral-7B keeps asking the salient *name* on 94% (regret-0 100% → 3%): it notices a column whose values are all equal (the extreme edit), but when name and values disagree it follows the name. *Anonymised*: names become `attr_i`. Qwen2.5-14B is unmoved (regret 0.103 before and after); DeepSeek-7B collapses (regret-0 71% → 36%); for Yi-1.5-9B 82% of wrong draws still land on the column it favours table-free, so its default is tied to the values' shape as much as to the name.

Natural tasks. On 70 unedited tables in which a strictly better attribute than the salient one exists, regret-0 collapses relative to 71 matched tables where the salient attribute is optimal: Mistral-7B 100% → 25.7%, GLM-4-9B 96.2% → 14.4%, Qwen2.5-14B 90.3% → 62.1%. Per source in Table 10; 27 of the 70 are SGD Movies.

A table-free default. Asked which attribute they would ask with *no* records shown, models answer sharply. On natural data, DeepSeek-7B's wrong draws land on its top table-free wrong attribute 94.5% [89.4, 98.8] of the time, Yi-1.5-9B's 84.7%, Gemma-2-9B's 85.4%, against a 30–44% uniform baseline (Appendix Fig. 5, Table 11); GLM-4-9B (48%) and Qwen2.5-7B (29%) do not fit. Some models carry a table-independent default that overrides the table; we do not claim this is the general mechanism.

	exact distinct-value counts given			instruction to count		
	mean regret	Δ [95% CI]	mild edit: asks it	mean regret	Δ [95% CI]	fixed / broken
Qwen2.5-32B	0.013 → 0.000	-0.013 [-0.027, -0.002]	3.7% → 12.0%	0.013 → 0.043	+0.030 [+0.011, +0.052]*	0 / 15
Qwen2.5-14B	0.022 → 0.042	+0.020 [+0.003, +0.041]	8.8% → 12.3%	0.022 → 0.026	+0.004 [-0.016, +0.024]	2 / 7
Mistral-7B	0.030 → 0.036	+0.007 [-0.008, +0.022]	82.5% → 67.8%	0.029 → 0.072	+0.042 [+0.014, +0.076]*	1 / 12
Qwen2.5-7B	0.058 → 0.073	+0.015 [-0.026, +0.056]	34.6% → 1.6%	0.058 → 0.039	-0.019 [-0.052, +0.015]	11 / 7
Llama-3.1-8B	0.048 → 0.023	-0.025 [-0.058, +0.002]	13.0% → 34.6%	0.048 → 0.141	+0.093 [+0.058, +0.130]*	2 / 61
Gemma-2-9B	0.070 → 0.093	+0.024 [+0.003, +0.051]	21.1% → 15.8%	0.070 → 0.014	-0.056 [-0.096, -0.022]*	11 / 2
GLM-4-9B	0.147 → 0.064	-0.083 [-0.122, -0.049]	77.6% → 72.2%	0.156 → 0.334	+0.178 [+0.087, +0.279]*	15 / 20
Yi-1.5-9B	0.168 → 0.225	+0.057 [-0.020, +0.141]	84.7% → 78.1%	0.168 → 0.190	+0.022 [-0.033, +0.080]	13 / 29
DeepSeek-7B	0.464 → 0.474	+0.010 [-0.065, +0.089]	72.8% → 74.4%	0.470 → 0.478	+0.009 [-0.074, +0.095]	15 / 22

Table 2: Two obvious repairs. Left: one line of tool output with the exact number of distinct values of every attribute is added to the prompt (choosing the largest is optimal on every task). Right: the model is told to count distinct values and prefer the largest; "fixed / broken" counts tasks that become / stop being fully optimal. Each comparison uses the tasks on which both runs give a parsable answer, so a model's baseline can differ between the two blocks (GLM-4-9B, which loses tables to unparsable draws, by up to 0.01 bit). *: interval excludes zero.

4.4 KNOWING THE FACTS, OR THE GOAL, IS NOT ENOUGH

If models merely miscount, two obvious repairs should work: hand them the counts, or tell them to count (Table 2). If they merely optimise something else, telling them the goal should.

Exact counts. We append one line of tool output with the exact number of distinct values of every attribute; on every scored table the attribute with most values is EIG-optimal (Appendix A), so the line is an exact oracle here. Only two of nine models improve significantly (GLM-4-9B -0.083 bit; Qwen2.5-32B -0.013, reaching 0.000 bit); two get significantly worse (Qwen2.5-14B $+0.020$, Gemma-2-9B $+0.024$); the remaining five show no significant change. DeepSeek-7B is unchanged (0.464 → 0.474) and on the mild counterfactual still asks the edited attribute 74.4% of the time with the counts in front of it.

An instruction to count. Telling the model to count the distinct values of each attribute and prefer the largest is less reliable still (Table 2, right; Appendix Fig. 3). It significantly helps one model (Gemma-2-9B 0.070 → 0.014 bit, 11 tables fixed and 2 broken), leaves four unchanged (Qwen2.5-14B, Qwen2.5-7B, Yi-1.5-9B, DeepSeek-7B) and makes four significantly worse: Qwen2.5-32B 0.013 → 0.043 (0 tables fixed, 15 broken), GLM-4-9B 0.156 → 0.334, Llama-3.1-8B 0.048 → 0.141, Mistral-7B 0.029 → 0.072. The damage falls on tasks the model had already solved: of the tables answered optimally in every draw without the instruction, it breaks 61 of 158 for Llama-3.1-8B and 20 of 96 for GLM-4-9B, but 2 of 155 for Gemma-2-9B.

Stating the goal. Our prompt says that no attribute is the correct one. A variant states the objective ("leave the fewest records still possible, on average", which on every scored table selects the EIG-optimal attribute; Appendix A), says the customer knows every attribute of their record, and suggests no procedure (Table 21; the 32B in its 4-bit activation-aware weight quantisation (AWQ) build, Lin et al., 2024). Seven of nine models, and GPT-4o, get significantly *worse* on the unmodified benchmark (Qwen2.5-14B 0.022 → 0.149 bit, 0 tables fixed, 25 broken; Llama-3.1-8B 0.048 → 0.320) and eight of nine on the second sample; only Qwen2.5-7B improves there (0.151 → 0.103). For Llama-3.1-8B and DeepSeek-7B part of the loss is format (12% and 25% unparsable draws).

No model gains from more than one repair.

Two case studies. DeepSeek-7B ignores the counting instruction (20 of 20 sampled completions are bare JavaScript Object Notation (JSON), the output format the prompt asks for). Llama-3.1-8B reasons, miscounts, and also overrides a correct count on grounds of naturalness ("item_names has the most distinct values … However, asking about the items might not be the most natural question"), as the instruction's "unless it is clearly not a natural thing to ask" licenses. Other models were not inspected.

4.5 The cost of a poor first question

Five models keep asking, one attribute per round, on 87 tables on which the first model we studied had chosen a non-optimal attribute (a hard subset), the candidates being filtered after each round on the target's true value, with no simulated user (Table 26). Had a pipeline committed after one question, choosing uniformly among the candidates still consistent, a first question with regret $>$ 0.8 bit would raise the probability of a wrong record from 12–26% to 50–64% (four models with $\geq$ 15 such tables; within each model the intervals do not overlap), and the questions needed to resolve rise from 1.2–1.7 to 1.7–3.2. Follow-up turns resolve 90–100% of tables, but only 35–67% in the minimum number of rounds (Appendix Fig. 4).

5 Discussion

What regret does and does not show. Our findings rest on two footings. Where a column has been flattened, asking about it has zero value under any utility increasing in information, so a model that keeps asking trades information for nothing: a failure whatever the user's cost of answering. Where a column is merely not the most informative, regret is the exact information given up under this uniform-prior EIG metric—not a bound on lost user-aware utility—and it is the true cost of the question only if the user could have answered the better one. Findings (1) and (2) stand on the first footing; finding (3) lives on the second. On SGD Movies the most informative attribute is the exact ticket price, which a caller may not remember, and the strongest models ask the format or genre instead (Table 25); stating the goal does not move them (§4.4), and one model's completions set the most discriminating attribute aside as "not the most natural question". Whether that is judgment or habit, regret cannot tell; scoring questions by information net of user cost (Rao & Daumé III, 2018) is the natural next step.

For agent builders. Both failures are invisible in the transcript: a question about a column that no longer separates anything reads like a good one, and a follow-up turn usually repairs it (§4.5), so eventual-resolution metrics can conceal additional interaction cost. When the candidate set is closed, which attribute discriminates is a computation, not a judgment; the judgment to leave to the model is which discriminating attribute a user can answer.

6 Limitations

Regret is information lost by one question, not task success (§4.5); linking it to end-to-end success needs a user simulator, itself an unreliable proxy (Seshadri et al., 2026; Shim et al., 2026), and only τ-bench provides one, with 17 benchmark tables. We study nine open-weight instruction-tuned models of 7–32B parameters in full precision (bfloat16, *bf16* in Table 16), plus one frontier API model in the appendix. The 4-bit AWQ build of the 32B model has 1.6$\times$ its regret (Table 16). The setting is one clarifying question over at most eight retrieved records, under a uniform prior, with a user who knows every listed attribute; row and column order and record ids were not permuted. Evidence for a table-independent default is causal for one model and partial for two (§4.3). Magnitudes vary with prompt wording and decoding temperature (Appendix A); the sign does not.

AI use statement

In the data construction phase, Claude was utilized to assist in the deterministic edit of the dataset. Regarding the manuscript and experimental implementation, Gemini was used solely for language polishing, grammatical correction, and assistance with basic coding and organizing the supplement package. The authors verified all AI-generated content and assume full responsibility for the final outcomes.

Reproducibility statement

All candidate sets come from public data (SGD, MultiWOZ, τ-bench, Olist; the banking tables from the `dev/tau3` branch of the public τ^2-bench repository, commit `17e07b1`). §3.1 and Tables 13–14 give every construction step with counts; both samples are drawn by seeded scripts. The

supplementary material contains the task files, the prompts, the raw outputs of every model call, the deterministic scoring code that regenerates every table and figure, a script that checks every number in the text against the generated tables, model revisions and decoding parameters, and a single definitions file for regret, information-tied tasks, near-ID attributes and the unit of analysis.

Ethics statement

The study uses public dialogue and e-commerce datasets under their licences; the Olist records are anonymised by the data provider and no personal data is added. No human subjects were involved. The work measures a failure of language-model agents and proposes no deployment.

References

01.AI. Yi: Open foundation models by 01.AI. *arXiv preprint arXiv:2403.04652*, 2024. reports Yi-6B/34B; we run the later Yi-1.5-9B-Chat release of the same authors.

Mohammad Aliannejadi, Hamed Zamani, Fabio Crestani, and W. Bruce Croft. Asking clarifying questions in open-domain information-seeking conversations. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pp. 475–484. ACM, 2019.

Joris Baan, Wilker Aziz, Barbara Plank, and Raquel Fernández. Clarify, abstain or answer? strategising in conversation with belief-augmented generation. In *Proceedings of the International Natural Language Generation Conference (INLG)*, 2026. arXiv:2605.25831.

Victor Barres, Honghua Dong, Soham Ray, Xujie Si, and Karthik Narasimhan. τ^2-bench: Evaluating conversational agents in a dual-control environment, 2025. arXiv:2506.07982.

Leonardo Bertolazzi, Davide Mazzaccara, Filippo Merlo, and Raffaella Bernardi. ChatGPT's information seeking strategy: Insights from the 20-questions game. In *Proceedings of the 16th International Natural Language Generation Conference (INLG)*, pp. 153–162, Prague, Czechia, 2023. Association for Computational Linguistics.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. MultiWOZ – a large-scale multi-domain Wizard-of-Oz dataset for task-oriented dialogue modelling. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018.

Deepro Choudhury, Sinead Williamson, Adam Goliński, Ning Miao, Freddie Bickford Smith, Michael Kirchhof, Yizhe Zhang, and Tom Rainforth. BED-LLM: Intelligent information gathering with LLMs and Bayesian experimental design. In *International Conference on Learning Representations (ICLR)*, 2026. arXiv:2508.21184.

Konstantina Christakopoulou, Filip Radlinski, and Katja Hofmann. Towards conversational recommender systems. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 815–824. ACM, 2016.

DeepSeek-AI. DeepSeek LLM: Scaling open-source language models with longtermism. *arXiv preprint arXiv:2401.02954*, 2024.

Mengyi Deng, Zhiwei Li, Xin Li, Tingyu Zhu, Ying Zhao, Zhijiang Guo, and Wei Wang. Uncertainty-aware clarification in LLM agents with information gain. In *International Conference on Machine Learning (ICML)*, 2026a. arXiv:2606.03135.

Mingyi Deng, Lijun Huang, Yani Fan, Fanqi Kong, Jiayi Zhang, Fashen Ren, Jinyi Bai, Fuzhen Yang, Dayi Miao, Zhaoyang Yu, Yifan Wu, Yanfei Zhang, Fengwei Teng, Yingjia Wan, Song Hu, Yude Li, Xin Jin, Conghao Hu, Haoyu Li, Qirui Fu, Tai Zhong, Xinyu Wang, Xiangru Tang, Nan Tang, Chenglin Wu, and Yuyu Luo. InteractComp: Evaluating search agents with ambiguous queries. In *International Conference on Machine Learning (ICML)*, 2026b. arXiv:2510.24668.

Michael C. Frank and Noah D. Goodman. Predicting pragmatic reasoning in language games. *Science*, 336(6084):998–998, 2012.

Gemma Team. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024.

GLM Team. ChatGLM: A family of large language models from GLM-130B to GLM-4 all tools. *arXiv preprint arXiv:2406.12793*, 2024.

Gabriel Grand, Valerio Pepe, Jacob Andreas, and Joshua B. Tenenbaum. Shoot first, ask questions later? building rational agents that explore and act like people. In *International Conference on Learning Representations (ICLR)*, 2026. arXiv:2510.20886.

Anmol Gulati, Hariom Gupta, Elias Lumer, Sahil Sen, and Vamse Kumar Subbiah. Ask early, ask late, ask right: When does clarification timing matter for long-horizon agents? *arXiv preprint arXiv:2605.07937*, 2026.

Sarthak Harne, Natwar Modani, Debabrata Mahapatra, and Shubham Agarwal. Dialogue to discovery: Attribute-aware preference elicitation for conversational product search assistants. *arXiv preprint arXiv:2606.24194*, 2026.

Jakob Hartmann, James Harvey, Jhonathan Navott, Erik Y. Wang, Luckeciano C. Melo, Flaviu Cipcigan, Cheng Zhang, and Alessandro Abate. Amortising Bayesian experimental design for sequential information gathering in LLMs. *arXiv preprint arXiv:2607.03426*, 2026.

Zhiyuan Hu, Chumin Liu, Xidong Feng, Yilun Zhao, See-Kiong Ng, Anh Tuan Luu, Junxian He, Pang Wei Koh, and Bryan Hooi. Uncertainty of thoughts: Uncertainty-aware planning enhances information seeking in large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. arXiv:2402.03271.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7B. *arXiv preprint arXiv:2310.06825*, 2023. reports v0.1; we run the later Mistral-7B-Instruct-v0.3 release.

Katarzyna Kobalczyk, Nicolas Astorga, Tennison Liu, and Mihaela van der Schaar. Active task disambiguation with LLMs. In *International Conference on Learning Representations (ICLR)*, 2025. arXiv:2502.04485.

Wenqiang Lei, Xiangnan He, Yisong Miao, Qingyun Wu, Richang Hong, Min-Yen Kan, and Tat-Seng Chua. Estimation-action-reflection: Towards deep interaction between conversational and recommender systems. In *Proceedings of the 13th International Conference on Web Search and Data Mining (WSDM)*, pp. 304–312. ACM, 2020.

Miao Li, Zhipeng Chen, and Ji Wu. Improving the probabilistic framework for representing dialogue systems with user response model. In *Proceedings of Interspeech 2016*, pp. 2701–2705, 2016. doi: 10.21437/Interspeech.2016-810.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. AWQ: Activation-aware weight quantization for on-device LLM compression and acceleration. In *Proceedings of Machine Learning and Systems (MLSys)*, 2024. arXiv:2306.00978.

Dennis V. Lindley. On a measure of the information provided by an experiment. *The Annals of Mathematical Statistics*, 27(4):986–1005, 1956.

Llama Team, AI @ Meta. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Bolei Ma and Yusuke Miyao. The imperfective paradox in large language models. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2026. arXiv:2601.09373.

Davide Mazzaccara, Alberto Testoni, and Raffaella Bernardi. Learning to ask informative questions: Enhancing LLMs with preference optimization and expected information gain. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 5064–5074, Miami, Florida, USA, 2024. Association for Computational Linguistics.

Olist. Brazilian e-commerce public dataset by Olist. Kaggle dataset, 2018.

OpenAI. GPT-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

Kun Qian, Satwik Kottur, Ahmad Beirami, Shahin Shayandeh, Paul Crook, Alborz Geramifard, Zhou Yu, and Chinnadhurai Sankar. Database search results disambiguation for task-oriented dialog systems. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, pp. 1158–1173, 2022.

J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, 1986.

Qwen Team. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2025.

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

Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.

Preethi Seshadri, Samuel Cahyawijaya, Ayomide Odumakinde, Sameer Singh, and Seraphina Goldfarb-Tarrant. Lost in simulation: LLM-simulated users are unreliable proxies for human users in agentic evaluations, 2026. arXiv:2601.17087.

Jeonghoon Shim, Woojung Song, Cheyon Jin, Seungwon Kook, and Yohan Jo. Non-collaborative user simulators for tool agents. In *International Conference on Learning Representations (ICLR)*, 2026. arXiv:2509.23124.

Yuan Sui, Mengyu Zhou, Mingjie Zhou, Shi Han, and Dongmei Zhang. Table meets LLM: Can large language models understand structured table data? a benchmark and empirical study. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining (WSDM)*, pp. 645–654. ACM, 2024.

Manan Suri, Puneet Mathur, Nedim Lipka, Franck Dernoncourt, Ryan A. Rossi, and Dinesh Manocha. Structured uncertainty guided clarification for LLM agents. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 40811–40838, 2026.

Yiling Tao, Shihan Deng, Meiling Tao, Pengzhi Wei, Zhichao Hu, and Zhihao Zhu. When search agents should ask: DiscoBench for clarification-aware deep search, 2026. arXiv:2606.27669.

Yuqing Yang, Qi Zhu, Zhen Han, Boran Han, Zhengyuan Shen, Shuai Wang, Vassilis N. Ioannidis, and Huzefa Rangwala. When LLMs read tables carelessly: Measuring and reducing data referencing errors. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 16734–16752, San Diego, California, USA, 2026. Association for Computational Linguistics.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. τ-bench: A benchmark for tool-agent-user interaction in real-world domains. *arXiv preprint arXiv:2406.12045*, 2024.

Hamed Zamani, Susan Dumais, Nick Craswell, Paul Bennett, and Gord Lueck. Generating clarifying questions for information retrieval. In *Proceedings of The Web Conference 2020 (WWW)*, pp. 418–428. ACM, 2020.

Michael J. Q. Zhang, W. Bradley Knox, and Eunsol Choi. Modeling future conversation turns to teach LLMs to ask clarifying questions. In *International Conference on Learning Representations (ICLR)*, 2025. arXiv:2410.13788.

Jiale Zhao, Ke Fang, and Lu Cheng. When and what to ask: AskBench and rubric-guided RLVR for LLM clarification, 2026. arXiv:2602.11199.

Deyu Zou, Yongqiang Chen, Fan Feng, Mufei Li, Pan Li, Yu Gong, and James Cheng. On information self-locking in reinforcement learning for active reasoning of LLM agents. In *International Conference on Machine Learning (ICML)*, 2026. arXiv:2603.12109.

	original	terse	"fewest questions"	markdown table	Δ terse [95% CI]	Δ fewest	Δ markdown
Qwen2.5-32B	0.013	0.046	0.149	0.003	+0.033 [+0.005, +0.066]*	+0.137 [+0.088, +0.190]*	-0.010 [-0.024, +0.003]
Qwen2.5-14B	0.022	0.045	0.105	0.036	+0.023 [-0.005, +0.054]	+0.083 [+0.046, +0.129]*	+0.014 [-0.001, +0.031]
Mistral-7B	0.030	0.082	0.032	0.078	+0.052 [+0.024, +0.082]*	+0.003 [-0.013, +0.018]	+0.048 [+0.020, +0.080]*
Qwen2.5-7B	0.058	0.115	0.109	0.053	+0.057 [+0.012, +0.109]*	+0.051 [+0.009, +0.098]*	-0.005 [-0.029, +0.018]
Llama-3.1-8B	0.048	0.061	0.190	0.085	+0.014 [-0.016, +0.042]	+0.142 [+0.094, +0.193]*	+0.037 [+0.011, +0.063]*
Gemma-2-9B	0.070	0.079	0.178	0.123	+0.009 [-0.037, +0.055]	+0.109 [+0.058, +0.162]*	+0.053 [+0.019, +0.088]*
GLM-4-9B	0.147	0.370	0.380	0.302	+0.223 [+0.151, +0.297]*	+0.233 [+0.156, +0.316]*	+0.155 [+0.089, +0.223]*
Yi-1.5-9B	0.168	0.200	0.227	0.158	+0.032 [-0.034, +0.095]	+0.059 [-0.007, +0.126]	-0.010 [-0.063, +0.046]
DeepSeek-7B	0.464	0.497	0.526	0.414	+0.033 [-0.020, +0.087]	+0.062 [-0.004, +0.129]	-0.050 [-0.107, +0.009]

Table 3: Mean regret (bit) under four prompt wordings, same tables within a row: the original; terse with no persona; another persona with the goal stated as "as few questions as possible"; the original with records rendered as a markdown table. Δ: paired change relative to the original, task-level bootstrap; *: interval excludes zero. No wording is significantly lower than the original.

	extreme edit (column flattened, EIG 0)			mild edit (half collide)	
	asks it, before	asks it, after [95% CI]	regret-0	asks it, after (seed 0 / 1)	regret-0 (after: seed 0 / 1)
Qwen2.5-32B	74.6%	0.0% [0.0, 0.0]	91.5% → 96.9%	3.7% / 3.2%	90.3% → 89.2% / 88.8%
Qwen2.5-14B	71.2%	0.0% [0.0, 0.0]	93.0% → 89.2%	8.8% / 5.4%	93.8% → 76.1% / 80.4%
Mistral-7B	97.5%	0.9% [0.0, 2.8]	100.0% → 93.2%	82.5% / 75.9%	100.0% → 17.2% / 24.1%
Qwen2.5-7B	52.9%	0.0% [0.0, 0.0]	85.8% → 95.4%	34.6% / 30.7%	86.7% → 62.2% / 65.8%
Llama-3.1-8B	69.7%	0.0% [0.0, 0.0]	86.2% → 93.1%	13.0% / 17.5%	84.2% → 78.4% / 73.0%
Gemma-2-9B	66.8%	0.0% [0.0, 0.0]	81.5% → 84.9%	21.1% / 19.1%	78.9% → 61.4% / 65.1%
GLM-4-9B	90.1%	23.3% [17.0, 29.9]	95.2% → 47.0%	77.6% / 81.9%	95.4% → 9.4% / 9.5%
Yi-1.5-9B	82.2%	31.8% [22.3, 42.2]	83.7% → 44.2%	84.7% / 85.3%	88.4% → 1.2% / 1.8%
DeepSeek-7B	69.2%	62.0% [50.3, 73.2]	75.1% → 7.7%	72.8% / 73.0%	80.4% → 7.0% / 7.7%

Table 4: Paired single-column counterfactuals on near-ID-free sources ($N \geq 3$). "Asks it" = share of draws asking the edited (salient) attribute; uniform-random baselines 29.9% (extreme) and 31.2% (mild). Regret-0 is shown before → after the edit (for the mild edit, after = seed 0 / seed 1).

A ADDITIONAL TABLES

All tables in this paper are generated from the raw model outputs by the released scoring pipeline. Table 23 lists every task set, its selection rule and its size.

Reverse edit (69 tables): the salient attribute is made optimal. Every model asks it more often (e.g. Mistral-7B 71% → 97%, Qwen2.5-14B 16% → 60%) except Qwen2.5-7B (22.9% → 25.9%) and DeepSeek-7B (45.2% → 45.5%), which barely move.

Robustness of the construction. *Models.* We run Qwen2.5-32B/14B/7B-Instruct, Llama-3.1-8B-Instruct, Gemma-2-9B-it, GLM-4-9B-Chat, Mistral-7B-Instruct-v0.3, Yi-1.5-9B-Chat and DeepSeek-LLM-7B-Chat; the exact revision of each is listed in the supplementary material. The Mistral and Yi reports describe earlier releases (v0.1; Yi-6B/34B) than the checkpoints we run. *Question vs. declared attribute.* The scorers read the attribute field of the model's JSON. On the goal-stated runs, whose full call logs we kept (11,680 calls), a match of the attribute's name, a word stem of it or one of its table values finds the declared attribute in the question in 90–100% of parsable calls per model; the remainder are paraphrases the matcher misses ("What city did you depart from?" for `origin`). We found no question that asks about a different attribute than the one declared. 84% of the questions are open ("what was the price?"), whose answer partitions the candidates exactly as the attribute does; 9% are yes/no or A-or-B, almost all on two-valued attributes, for which the two forms coincide. For the remaining 7%, which our form classifier does not label, we do not claim that the partition induced by the phrased question equals the partition of the declared attribute; the scored quantity is the declared attribute throughout, and a question of a more restrictive form than its attribute would make our regret figures optimistic, not pessimistic. *Count and goal oracles.* On all 171 unmodified benchmark tables and all 251 second-sample tables, every attribute with the most distinct values is EIG-optimal, and so is every attribute that minimises the expected number of records left; the two repairs of §4.4 therefore hand the model, or state, the EIG objective on every scored table. This coincidence is a property of our tables, not an identity: with $N = 7$ the partitions $(4, 1, 1, 1)$ and $(3, 2, 2)$ carry 1.664 and 1.557 bit but leave 2.714 and 2.429 records on average, so the higher-EIG question is the one that leaves *more* records (the many-valued bias of Quinlan, 1986). We verified the coincidence on every scored table rather than assuming it. *Unparsable outputs.*

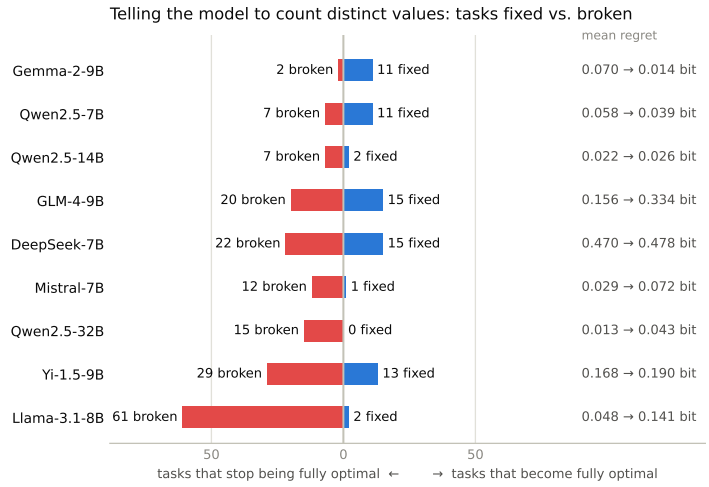


Figure 3: The instruction to count distinct values, per model: tasks that become fully optimal (right) against tasks that stop being fully optimal (left), with the change in mean regret. The counting instruction improves some models and degrades others.

source	attribute	all-distinct rate	share of picks
banking_credit_card_accounts	card_type	100% (5)	91%
banking_credit_card_accounts	date_of_account_open	100% (5)	9%
sgd_Flights_1	airlines	100% (228)	75%
sgd_Flights_1	inbound_departure_time	91% (89)	0%
sgd_Flights_1	outbound_departure_time	99% (228)	14%
sgd_Flights_2	airlines	100% (64)	41%
sgd_Flights_2	inbound_arrival_time	91% (23)	6%
sgd_Flights_2	inbound_departure_time	100% (23)	0%
sgd_Flights_2	outbound_departure_time	97% (64)	38%
sgd_Media_1	directed_by	100% (31)	11%
sgd_Media_1	title	100% (31)	89%
tau_bench_airline	destination	100% (5)	9%
tau_bench_airline	flight_dates	100% (5)	49%
tau_bench_airline	flight_numbers	100% (5)	0%
tau_bench_retail	item_names	100% (36)	73%

Table 5: Attributes flagged by the near-identifier rule (all candidates differ on $\geq 90\%$ of a source's $N \geq 3$ sets).

Draws whose attribute cannot be parsed are left out of regret; counting them as errors materially affects only GLM-4-9B (0.129 $\rightarrow$ 0.161–0.185 bit). *Tie threshold.* Moving it over $\{0, 0.1, 0.2, 0.3\}$ changes no model's mean regret by more than 0.006 bit and no rank (Table 12). *Source exclusion.* Under three policies applied to the original, unmodified tasks—all 17 sources, MultiWOZ excluded, only sources without a near-ID attribute—DeepSeek-7B and Yi-1.5-9B are last every time, while the order among the top models, which are statistically indistinguishable (§4.1), changes; leaving near-ID sources in hands every model free points and shrinks the effect (Table 7). *Source weights.* Re-weighting the benchmark so that no source counts for more than 15 tasks, or so that all sources count equally, swaps at most one pair of adjacent, statistically indistinguishable models and moves no model's mean regret by more than 0.03 bit except DeepSeek-7B (0.406 $\rightarrow$ 0.322 and 0.301), whose errors are concentrated in the largest source.

Decoding temperature. The main experiments sample at T=0.3 with k=5. Re-running the unmodified tasks with greedy decoding (T=0) and at T=1.0 leaves every model's regret above zero and the top three and the last model in place; rank changes occur only among adjacent, statistically indistinguishable models (Table 15). Relative to T=0.3, greedy decoding changes two models significantly, in opposite directions (GLM-4-9B 0.148 $\rightarrow$ 0.099, DeepSeek-7B 0.467 $\rightarrow$ 0.514), and T=1.0 makes three models significantly worse (GLM-4-9B 0.312, Llama-3.1-8B 0.079,

threshold	attributes flagged	sources flagged	sources vs. 0.90
0.80	23	9	+attraction_db; +restaurant_db; +sgd_Events_2
0.90	15	6	same sources
0.95	13	6	same sources

Table 6: Near-identifier rule at other thresholds. At 0.95 the same six sources are flagged (two fewer columns), so the source-level exclusion is unchanged. At 0.80 three more sources are flagged; two of them are MultiWOZ sources, whose exclusion is covered by Table 7. Only the flags are recomputed here: scoring under another threshold would require rebuilding tasks and new model runs.

policy (original tasks only)	tasks	Qwen2.5-32B	Qwen2.5-14B	Mistral-7B	Qwen2.5-7B	Llama-3.1-8B	Gemma-2-9B	GLM-4-9B	Yi-1.5-9B	DeepSeek-7B
all 17 sources	247	0.027	0.029	0.036	0.049	0.070	0.051	0.099	0.149	0.369
MultiWOZ excluded	220	0.030	0.033	0.040	0.045	0.078	0.056	0.106	0.163	0.401
MultiWOZ and near-ID sources excluded	144	0.015	0.026	0.035	0.053	0.057	0.080	0.160	0.193	0.531

Table 7: Mean regret (bit) under three source-exclusion policies, before any source is rebuilt. DeepSeek-7B and Yi-1.5-9B are last under every policy, and the order among the statistically indistinguishable models in between changes. The policies are cumulative: the third also excludes MultiWOZ. The 247 tasks of the first row are the 292 sampled tasks minus the 7 information-tied ones (there are no others) and minus 38 tasks whose table is identical to another's. Removing the near-ID sources raises the regret of the weaker models and leaves the strongest flat or lower.

Qwen2.5-14B 0.032) and none better, so nonzero regret persists under greedy decoding and higher-temperature sampling.

Why a uniform draw from Olist yields almost only two-candidate tasks. Olist is by far the largest pool, but repeat purchases are rare in it: of 96,096 customers, 93,099 placed a single order, 2,745 placed two, 203 three and 49 four or more. A candidate set is one customer's orders, so 2,593 of the 2,797 pool tasks (92.7%) have $N{=}2$ and only 204 have $N \geq 3$. A uniform draw of 60 is expected to contain 4.4 tasks with $N \geq 3$; ours contains 3, and raising the cap would not change the proportion. With two candidates any attribute on which they differ is optimal, which is why the six stronger models have zero regret on Olist (Table 8) while the source still separates DeepSeek-7B, which asks attributes on which the two records agree. The harder Olist tables enter our study through the second sample (60 of its 151 eligible $N \geq 3$ tables; Table 19) and the natural-task contrast (Table 10).

	tasks	Qwen2.5-32B	Qwen2.5-14B	Mistral-7B	Qwen2.5-7B	Llama-3.1-8B	Gemma-2-9B	GLM-4-9B	Yi-1.5-9B	DeepSeek-7B
$N = 2$	108	0.000	0.000	0.009	0.009	0.000	0.000	0.108	0.193	0.539
$N = 3$-4	97	0.062	0.101	0.142	0.131	0.161	0.159	0.154	0.174	0.314
$N = \geq 5$	33	0.077	0.143	0.103	0.153	0.159	0.289	0.125	0.100	0.238
attraction_db	10	0.000	0.000	0.000	0.200	0.000	0.032	0.000	0.000	0.116
banking_accounts	10	0.000	0.000	0.000	0.067	0.013	0.000	0.200	0.032	0.762
hotel_db	7	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.105
olist	59	0.000	0.000	0.000	0.000	0.000	0.000	0.137	0.227	0.820
restaurant_db	10	0.000	0.000	0.000	0.020	0.000	0.000	0.112	0.100	0.100
sgd_Events_1	12	0.000	0.000	0.111	0.083	0.339	0.392	0.191	0.167	0.437
sgd_Events_2	14	0.000	0.000	0.049	0.000	0.081	0.000	0.252	0.014	0.121
sgd_Flights_1	25	0.027	0.081	0.144	0.096	0.184	0.212	0.096	0.144	0.242
sgd_Flights_2	25	0.183	0.302	0.183	0.183	0.183	0.183	0.073	0.183	0.203
sgd_Movies_1	18	0.120	0.186	0.166	0.291	0.159	0.340	0.274	0.227	0.325
sgd_Music_1	9	0.000	0.000	0.000	0.000	0.000	0.000	0.030	0.111	0.201
sgd_Music_2	13	0.000	0.031	0.000	0.057	0.000	0.057	0.048	0.077	0.190
sgd_Restaurants_1	9	0.000	0.000	0.000	0.000	0.000	0.000	0.148	0.637	0.382
tau_bench_airline	5	0.133	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
tau_bench_retail	12	0.041	0.097	0.405	0.158	0.294	0.266	0.193	0.338	0.506

Table 8: Mean regret (bit) by number of candidates and by source. Flights, retail and airline tasks are rebuilt without their near-ID attributes.

	Qwen2.5-32B	Qwen2.5-14B	Mistral-7B	Qwen2.5-7B	Llama-3.1-8B	Gemma-2-9B	GLM-4-9B	Yi-1.5-9B	DeepSeek-7B
Qwen2.5-32B	–								
Qwen2.5-14B	0.386	–							
Mistral-7B	**0.013**	1.000	–						
Qwen2.5-7B	**0.025**	1.000	1.000	–					
Llama-3.1-8B	**0.008**	0.840	1.000	1.000	–				
Gemma-2-9B	**0.004**	0.142	0.795	0.840	0.702	–			
GLM-4-9B	**0.004**	**0.010**	0.061	0.154	0.526	1.000	–		
Yi-1.5-9B	**0.004**	**0.004**	**0.004**	**0.004**	**0.004**	0.085	0.795	–	
DeepSeek-7B	**0.004**	**0.004**	**0.004**	**0.004**	**0.004**	**0.004**	**0.004**	**0.004**	–

Table 9: Holm-adjusted p-values (paired sign-flip permutation test on task-level mean regret; row model worse than column model). Bold: $p < 0.05$.

	regret-0, salient optimal	regret-0, better attribute exists	asks salient anyway	by source: Olist / Movies / Events
Qwen2.5-32B	87.6% [80.0, 94.9]	61.7% [50.3, 72.9]	20.4% [11.9, 30.0]	14% (36) / 26% (27) / 38% (6)
Qwen2.5-14B	90.3% [83.4, 96.2]	62.1% [50.7, 73.6]	15.7% [7.1, 24.3]	6% (36) / 33% (27) / 0% (6)
Mistral-7B	100.0% [100.0, 100.0]	25.7% [15.7, 36.0]	70.3% [59.7, 80.9]	56% (36) / 96% (27) / 50% (6)
Qwen2.5-7B	76.6% [66.8, 85.6]	71.7% [61.3, 80.9]	22.6% [13.7, 32.7]	3% (36) / 53% (27) / 9% (6)
Llama-3.1-8B	78.9% [70.1, 87.0]	52.9% [42.7, 63.6]	23.7% [15.6, 32.3]	28% (36) / 16% (27) / 32% (6)
Gemma-2-9B	80.0% [70.4, 88.2]	33.7% [23.1, 45.1]	34.9% [24.1, 45.7]	48% (36) / 25% (27) / 8% (6)
GLM-4-9B	96.2% [93.4, 98.6]	14.4% [7.5, 21.8]	70.1% [60.8, 78.8]	53% (36) / 97% (27) / 50% (6)
Yi-1.5-9B	74.1% [64.5, 83.7]	8.6% [2.9, 15.4]	57.4% [46.3, 68.0]	29% (36) / 96% (27) / 63% (6)
DeepSeek-7B	44.3% [33.5, 55.3]	11.0% [5.1, 17.7]	44.6% [33.1, 56.0]	0% (36) / 96% (27) / 87% (6)

Table 10: Natural (unedited) tables, $N \geq 3$, near-ID-free sources. The last column gives the share of draws asking the salient attribute per source (number of tables in parentheses); 27 of the tables are SGD Movies, where the salient attribute is `genre`. The three sources shown hold 69 of the 70 tables; the rest are from SGD Restaurants.

	wrong draws used / all	lands on top table-free wrong attribute	uniform baseline	same, when it is not the EIG runner-up
Qwen2.5-32B	15 / 15 (100%)	66.7% [0.0, 100.0]	50.0%	66.7% [0.0, 100.0] (n=15)
Qwen2.5-14B	15 / 15 (100%)	33.3% [0.0, 100.0]	50.0%	n=0, too few
Mistral-7B	20 / 20 (100%)	100.0% [100.0, 100.0]	50.0%	n=0, too few
Qwen2.5-7B	31 / 44 (70%)	29.0% [0.0, 90.0]	41.9%	0.0% [0.0, 0.0] (n=22)
Llama-3.1-8B	20 / 35 (57%)	30.0% [0.0, 71.4]	50.0%	n=1, too few
Gemma-2-9B	41 / 41 (100%)	85.4% [51.7, 100.0]	43.9%	85.7% [40.0, 100.0] (n=35)
GLM-4-9B	98 / 100 (98%)	48.0% [30.0, 65.5]	37.3%	34.6% [19.8, 52.4] (n=78)
Yi-1.5-9B	131 / 131 (100%)	84.7% [70.6, 96.4]	35.1%	80.0% [61.5, 95.1] (n=100)
DeepSeek-7B	344 / 355 (97%)	94.5% [89.4, 98.8]	29.5%	96.7% [91.5, 100.0] (n=274)

Table 11: Does the attribute a model favours when shown no records predict where its errors go? Natural data.

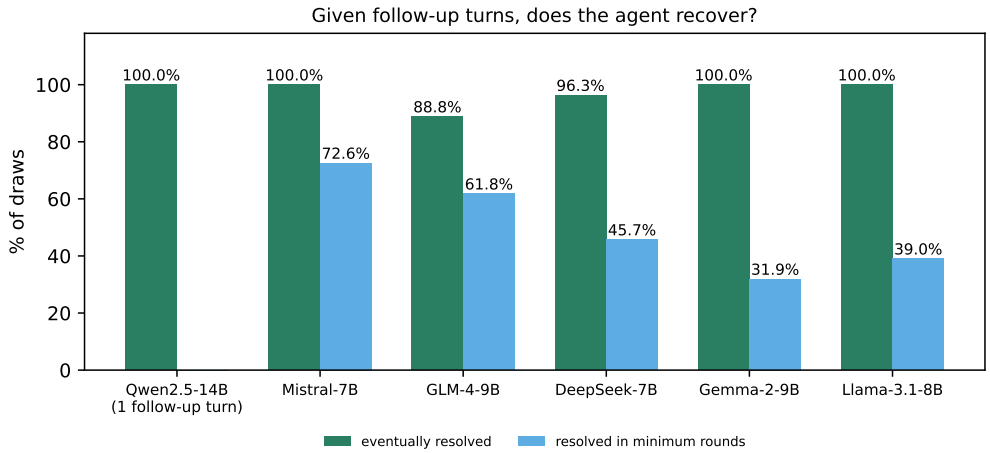


Figure 4: Follow-up turns hide the cost of a poor first question; they do not remove it.

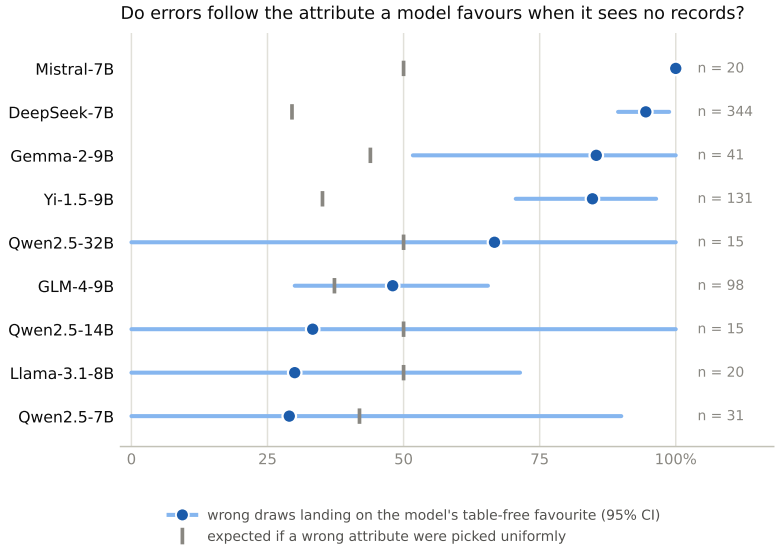


Figure 5: Share of wrong draws that land on the attribute a model favours when it is shown no records, with 95% task-level bootstrap intervals, against the share expected if a wrong attribute were picked uniformly (grey tick). The pattern is clear for DeepSeek-7B and Yi-1.5-9B, absent for GLM-4-9B, and undetermined for models with few errors (n = wrong draws used).

	exclude nothing	$\Delta < 0.1$	$\Delta < 0.2$ (used)	$\Delta < 0.3$	rank at each
Qwen2.5-32B	0.012	0.013	0.013	0.013	1 1 1 1
Qwen2.5-14B	0.021	0.022	0.022	0.022	2 2 2 2
Mistral-7B	0.031	0.029	0.029	0.029	3 3 3 3
Qwen2.5-7B	0.058	0.058	0.058	0.058	5 5 5 5
Llama-3.1-8B	0.047	0.048	0.048	0.048	4 4 4 4
Gemma-2-9B	0.070	0.070	0.070	0.070	6 6 6 6
GLM-4-9B	0.142	0.141	0.141	0.141	7 7 7 7
Yi-1.5-9B	0.165	0.168	0.168	0.168	8 8 8 8
DeepSeek-7B	0.458	0.464	0.464	0.464	9 9 9 9

Table 12: Mean regret (bit) when information-tied tasks are defined with different margins (unbiased-sample part of the benchmark).

step	tasks	sources	detail
source-capped sample	292	17	every model is run on all of them
− tasks of sources with a near-ID attribute	−106	−6	banking credit cards 11, SGD Flights_1 15, SGD Flights_2 15, SGD Media_1 15, τ-bench airline 7, τ-bench retail 43
− information-tied tasks	−7	0	of the 186 remaining tasks
= unmodified part of the benchmark	179	11	
+ tasks rebuilt without the near-ID attributes	+76	+4	SGD Flights_1 25, SGD Flights_2 25, τ-bench airline 7, τ-bench retail 19; 0 rebuilt tasks are information-tied; sources with no usable task left: banking credit cards, SGD Media_1
= scored tasks	255	15	
− tasks that show a table identical to another task (merged, draws pooled)	−17	0	unmodified 179 -¿ 171, rebuilt 76 -¿ 67; SGD Events_1 2, SGD Movies_1 1, SGD Music_2 2, SGD Restaurants_1 3, τ-bench airline 2, τ-bench retail 7
= benchmark (unique tables)	238	15	

Table 13: From the sample to the benchmark. Every model is run on all 292 sampled tasks and on the rebuilt tasks; scores use the 238 unique tables of the benchmark (the 171 unmodified ones for experiments that have no rebuilt counterpart). Information-tied: the best attribute that isolates the target is less than 0.2 bit more informative than the best one that does not.

source	pool	cap	sampled	near-ID attribute(s) removed	information-tied	in benchmark
olist	2797	60	60	–	1	59
sgd_Events_2	486	15	15	–	1	14
sgd_Flights_1	293	15	15	`airlines, inbound_departure_time, outbound_departure_time`	–	25 (rebuilt)
sgd_Movies_1	176	20	20	–	1	18
sgd_Events_1	86	15	15	–	1	12
sgd_Flights_2	78	15	15	`airlines, inbound_arrival_time, inbound_departure_time, outbound_departure_time`	–	25 (rebuilt)
tau_bench_retail	43	all	43	`item_names`	–	12 (rebuilt)
sgd_Media_1	37	15	15	`directed_by, title`	–	0 (rebuilt)
sgd_Restaurants_1	34	15	15	–	3	9
sgd_Music_2	32	15	15	–	0	13
restaurant_db	17	10	10	–	0	10
banking_credit_card_accounts	11	all	11	`card_type, date_of_account_open`	–	0 (rebuilt)
attraction_db	10	all	10	–	0	10
banking_accounts	10	all	10	–	0	10
sgd_Music_1	9	all	9	–	0	9
hotel_db	7	all	7	–	0	7
tau_bench_airline	7	all	7	`destination, flight_dates, flight_numbers`	–	5 (rebuilt)
total	4133		292		7	238

Table 14: Per-source construction. Pool: every closed-set task of the source with at least one attribute that isolates the target. Large pools are capped so that no source dominates (15 by default; 60 for Olist and 20 for SGD Movies, the two largest near-ID-free pools; 10 for the small MultiWOZ and banking pools; τ-bench is taken whole); within a source the draw is uniform (seed 123).

	tables	T=0 (greedy)	T=0.3 (main)	T=1.0	Δ T=0 vs 0.3 [95% CI]	Δ T=1.0 vs 0.3 [95% CI]	rank at each T
Qwen2.5-32B	171	0.013	0.013	0.013	+0.000 [+0.000, +0.000]	+0.000 [-0.007, +0.008]	1 / 1 / 1
Qwen2.5-14B	171	0.029	0.022	0.032	+0.007 [+0.000, +0.018]	**+0.011 [+0.002, +0.023]**	2 / 2 / 2
Mistral-7B	169	0.030	0.030	0.036	-0.000 [-0.012, +0.012]	+0.006 [-0.007, +0.019]	3 / 3 / 3
Qwen2.5-7B	171	0.052	0.058	0.055	-0.005 [-0.023, +0.009]	-0.003 [-0.015, +0.008]	4 / 5 / 4
Llama-3.1-8B	171	0.056	0.048	0.079	+0.008 [-0.000, +0.018]	**+0.031 [+0.014, +0.050]**	5 / 4 / 6
Gemma-2-9B	171	0.068	0.070	0.077	-0.002 [-0.006, +0.000]	+0.007 [-0.001, +0.018]	6 / 6 / 5
GLM-4-9B	162	0.099	0.148	0.312	**-0.049 [-0.075, -0.026]**	**+0.164 [+0.114, +0.213]**	7 / 7 / 8
Yi-1.5-9B	170	0.156	0.164	0.178	-0.008 [-0.018, +0.001]	+0.013 [-0.006, +0.033]	8 / 8 / 7
DeepSeek-7B	170	0.514	0.467	0.460	**+0.047 [+0.019, +0.078]**	-0.007 [-0.050, +0.035]	9 / 9 / 9

Table 15: Decoding sensitivity: mean regret (bit) with the original prompt on the unmodified tables, under greedy decoding (k=1), the main setting and T=1.0 (k=5). Bold: the task-level bootstrap interval of the change excludes zero.

experiment	tables	bf16	AWQ	AWQ − bf16 [95% CI]	regret-0 bf16 / AWQ
main benchmark (238 tables)	238	0.036	0.057	+0.021 [+0.004, +0.043]	93.7% / 91.2%
unmodified part (171 tables)	171	0.013	0.021	+0.008 [-0.002, +0.022]	97.7% / 97.1%
second sample (251 tables)	251	0.109	0.126	+0.017 [-0.005, +0.039]	86.5% / 83.7%
exact counts given (oc)	171	0.000	0.002	+0.002 [+0.000, +0.007]	100.0% / 99.4%
counting instruction (cot)	171	0.043	0.020	-0.023 [-0.042, -0.007]	88.9% / 91.8%
paraphrase p1	171	0.046	0.068	+0.022 [+0.005, +0.048]	93.0% / 89.5%
paraphrase p2	171	0.149	0.161	+0.012 [-0.013, +0.038]	81.3% / 79.5%
paraphrase p3	171	0.003	0.008	+0.005 [+0.000, +0.012]	99.4% / 97.7%
T = 0	170	0.013	0.022	+0.009 [-0.002, +0.026]	97.6% / 97.1%
T = 1.0	171	0.013	0.021	+0.008 [+0.001, +0.017]	97.1% / 95.9%
extreme edit (asks the edited attribute)	65	0.0%	0.0%	+0.0 pp [+0.0, +0.0]	—
mild edit, seed 0 (asks the edited attribute)	57	3.7%	10.8%	+7.0 pp [+1.4, +14.4]	—

Table 16: Quantisation check. Qwen2.5-32B-Instruct in bfloat16 (the build used in the paper) against the official 4-bit AWQ build of the same checkpoint on the same tasks, prompts, k and temperature; cells are mean regret (bit) unless marked pp (percentage points of draws asking the edited attribute). Intervals are task-level bootstrap.

condition			
Mean regret (bit) [95% CI]; rank among the nine open-weight models; their range			
main benchmark (238 tables)	0.101 [0.068, 0.136]	6 of 10	0.036–0.406
unmodified part (171 tables)	0.049 [0.026, 0.075]	5 of 10	0.013–0.464
second sample (251 tables)	0.187 [0.149, 0.226]	5 of 10	0.109–0.577
Share of draws asking the salient attribute, before → after the edit; regret-0 before → after; nine-model range after			
extreme edit (65 tables)	89.7% -¿ 0.0%	98.0% -¿ 92.9%	0.0%–62.0%
mild edit, seed 0 (57 tables)	88.2% -¿ 49.1%	97.7% -¿ 40.2%	3.7%–84.7%
mild edit, seed 1 (57 tables)	88.2% -¿ 45.4%	97.7% -¿ 45.8%	3.2%–85.3%
natural tasks: salient optimal (71) / better attribute exists (70)	regret-0 98.5% → 29.8%	asks salient 54.4%	–
Unmodified tables: mean regret; change vs the original prompt [95% CI]; regret-0			
original prompt, T = 0.3	0.049	—	92.0%
exact distinct-value counts given	0.007	-0.042 [-0.065, -0.022]	98.5%
paraphrase p1 (terse, no persona)	0.035	-0.014 [-0.042, +0.013]	95.4%
paraphrase p2 (other persona, fewest questions)	0.152	+0.103 [+0.059, +0.152]	81.8%
paraphrase p3 (markdown table)	0.074	+0.025 [+0.006, +0.046]	89.0%
T = 0 (greedy, k = 1)	0.063	+0.014 [+0.002, +0.029]	90.1%
T = 1.0	0.065	+0.016 [+0.005, +0.030]	89.6%
counting instruction, frozen parser (5 of 1460 draws unparsable, excluded)	0.000	-0.049 [-0.074, -0.027]	99.9%

Table 17: A frontier API model, gpt-4o-2024-08-06 (OpenAI API, fixed snapshot), run on the complete battery by a collaborator with the prompts, k, temperatures and parser unchanged; every condition has under 1% unparsable draws. It is scored exactly like the nine open-weight models but kept out of their tables. (An earlier run of the gpt-4o-2024-11-20 snapshot gave the same picture—benchmark regret 0.092, sixth of ten—but that snapshot writes set literals in its counting-instruction reasoning, which the frozen parser mistakes for the answer on 25% of draws; its files are released alongside.)

	mean regret (bit) [95% CI]	regret-0	non-opt. tables	mean regret, sources weighted equally
Qwen2.5-32B	0.109 [0.073, 0.148]	86.8%	13.5%	0.044 [0.028, 0.066]
Mistral-7B	0.125 [0.093, 0.157]	79.2%	21.2%	0.058 [0.040, 0.078]
Qwen2.5-14B	0.141 [0.101, 0.183]	81.3%	19.9%	0.073 [0.047, 0.104]
Qwen2.5-7B	0.151 [0.119, 0.187]	75.2%	29.5%	0.112 [0.068, 0.176]
Llama-3.1-8B	0.199 [0.154, 0.250]	76.9%	31.1%	0.113 [0.075, 0.162]
Gemma-2-9B	0.288 [0.233, 0.347]	66.8%	35.9%	0.140 [0.103, 0.192]
GLM-4-9B	0.293 [0.246, 0.346]	68.8%	45.8%	0.155 [0.109, 0.217]
Yi-1.5-9B	0.324 [0.264, 0.388]	64.9%	38.6%	0.178 [0.122, 0.246]
DeepSeek-7B	0.577 [0.501, 0.651]	47.3%	57.4%	0.495 [0.356, 0.653]
uniform random attribute	0.787	36.1%	–	–

Table 18: Second sample: 251 unmodified tables with $N \geq 3$ on which a wrong choice is possible (k=5, T=0.3, original prompt); the selection rules were fixed before any model was run on it. Intervals are task-level bootstrap. The last row is a uniform-random reference on the same tables.

	tables	most often optimal	Qwen2.5-32B	Mistral-7B	Qwen2.5-14B	Qwen2.5-7B	Llama-3.1-8B	Gemma-2-9B	GLM-4-9B	Yi-1.5-9B	DeepSeek-7B
$N = 3$	104		0.026	0.104	0.068	0.078	0.090	0.138	0.220	0.449	0.879
$N = 4$	61		0.115	0.100	0.147	0.151	0.166	0.269	0.357	0.206	0.392
$N = \geq 5$	86		0.204	0.169	0.225	0.240	0.356	0.483	0.336	0.258	0.344
olist	60	`purchase_month` 72%	0.031	0.084	0.042	0.040	0.064	0.083	0.257	0.752	1.341
sgd_Events_2	60	`date` 80%	0.000	0.020	0.064	0.137	0.126	0.099	0.444	0.062	0.138
sgd_Movies_1	57	`price` 88%	0.436	0.367	0.403	0.290	0.442	0.833	0.376	0.404	0.476
sgd_Events_1	37	`date` 51%	0.016	0.108	0.121	0.204	0.338	0.346	0.193	0.202	0.457
restaurant_db	12	`postcode` 100%	0.000	0.000	0.103	0.083	0.000	0.000	0.127	0.000	0.000
attraction_db	6	`postcode` 100%	0.000	0.000	0.000	0.225	0.000	0.053	0.000	0.000	0.661
sgd_Music_2	6	`album` 83%	0.000	0.000	0.067	0.078	0.000	0.124	0.038	0.000	0.244
sgd_Restaurants_1	4	`serves_alcohol` 50%	0.000	0.000	0.000	0.000	0.138	0.000	0.270	0.432	0.270
banking_accounts	3	`level` 100%	0.000	0.000	0.000	0.178	0.133	0.000	0.000	0.106	1.601
hotel_db	3	`postcode` 100%	0.000	0.044	0.000	0.000	0.000	0.000	0.000	0.000	0.044
sgd_Music_1	3	`year` 67%	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.216

Table 19: Second sample: mean regret (bit) by number of candidates and by source, with the attribute that is most often the optimal one in each source and the share of tables on which it is.

	Qwen2.5-32B	Mistral-7B	Qwen2.5-14B	Qwen2.5-7B	Llama-3.1-8B	Gemma-2-9B	GLM-4-9B	Yi-1.5-9B	DeepSeek-7B
Qwen2.5-32B	–								
Mistral-7B	1.000	–							
Qwen2.5-14B	0.905	1.000	–						
Qwen2.5-7B	0.661	1.000	1.000	–					
Llama-3.1-8B	**0.004**	**0.019**	0.075	0.533	–				
Gemma-2-9B	**0.004**	**0.004**	**0.004**	**0.004**	**0.004**	–			
GLM-4-9B	**0.004**	**0.004**	**0.004**	**0.004**	0.062	1.000	–		
Yi-1.5-9B	**0.004**	**0.004**	**0.004**	**0.004**	**0.017**	1.000	1.000	–	
DeepSeek-7B	**0.004**	**0.004**	**0.004**	**0.004**	**0.004**	**0.004**	**0.004**	**0.004**	–

Table 20: Second sample: Holm-adjusted p-values (paired sign-flip permutation test on task-level mean regret; row model worse than column model). Bold: $p < 0.05$.

	unmodified benchmark tables				second sample			
	regret	Δ [95% CI]	fixed / broken	unparsed	regret	Δ [95% CI]	fixed / broken	unparsed
Qwen2.5-32B†	0.021 → 0.161	+0.141 [+0.088, +0.201]*	0 / 26	0.0%	0.126 → 0.396	+0.270 [+0.218, +0.323]*	2 / 85	0.0%
Qwen2.5-14B	0.022 → 0.149	+0.127 [+0.079, +0.180]*	0 / 25	0.0%	0.141 → 0.357	+0.216 [+0.165, +0.271]*	2 / 56	0.0%
Mistral-7B	0.029 → 0.033	+0.003 [-0.011, +0.017]	1 / 4	0.0%	0.125 → 0.150	+0.025 [+0.001, +0.053]*	5 / 13	0.0%
Qwen2.5-7B	0.058 → 0.073	+0.015 [-0.027, +0.057]	12 / 11	0.0%	0.151 → 0.103	-0.048 [-0.087, -0.007]*	47 / 17	0.0%
Llama-3.1-8B	0.048 → 0.320	+0.272 [+0.208, +0.342]*	0 / 78	7.5%	0.200 → 0.646	+0.445 [+0.353, +0.544]*	5 / 96	12.0%
Gemma-2-9B	0.070 → 0.296	+0.227 [+0.159, +0.298]*	2 / 46	0.0%	0.288 → 0.482	+0.194 [+0.128, +0.266]*	7 / 46	0.0%
GLM-4-9B	0.142 → 0.287	+0.144 [+0.087, +0.204]*	10 / 38	8.7%	0.293 → 0.562	+0.269 [+0.210, +0.333]*	11 / 75	4.0%
Yi-1.5-9B	0.168 → 0.441	+0.273 [+0.210, +0.340]*	2 / 48	0.0%	0.324 → 0.494	+0.170 [+0.104, +0.239]*	10 / 42	0.0%
DeepSeek-7B	0.418 → 0.488	+0.070 [+0.017, +0.127]*	8 / 12	23.1%	0.370 → 0.606	+0.235 [+0.163, +0.308]*	10 / 48	24.5%
GPT-4o‡	0.049 → 0.133	+0.084 [+0.044, +0.126]*	1 / 13	0.0%	0.187 → 0.337	+0.150 [+0.110, +0.194]*	1 / 41	0.0%

Table 21: Stating the objective. Mean regret (bit) under the original prompt → under a prompt that states the goal ("the attribute whose answer will, on average, leave the fewest records still possible"), tells the model that the customer knows every attribute of their record, drops "there is no single correct attribute" and suggests no procedure (Appendix B); paired within a model on the tables it answered parsably under both prompts (this is why a model's baseline can differ from Table 1: DeepSeek-7B loses 16 and 55 tables to unparsable draws), task-level bootstrap. *: interval excludes zero. Fixed / broken: tables that become / stop being optimal in every draw. Unparsed: share of draws under the goal-stated prompt whose attribute could not be parsed. †: the official 4-bit AWQ build of the 32B checkpoint (Table 16); the bfloat16 model was not run under this prompt. ‡: the frontier API model of Table 17, outside the nine-model comparison.

	column-name swap (135 tables)				names anonymised (150 tables)	
	asks salient name, before	after [95% CI]	asks partner name, after	regret-0 before → after	regret-0 before → after [95% CI]	wrong draws on table-free default, before → after
Qwen2.5-14B	79.0%	27.9% [20.7, 35.7]	39.1% [31.2, 47.8]	85.8% → 54.6%	87.3% → 86.7% [81.3, 91.7]	26.9% (208) → 45.0% (100)
Mistral-7B	99.6%	93.8% [89.6, 97.3]	2.5% [0.3, 5.3]	100.0% → 3.3%	100.0% → 91.1% [86.7, 95.3]	– (0) → 70.1% (67)
Qwen2.5-7B	54.1%	33.8% [26.1, 41.6]	7.4% [3.7, 11.9]	67.9% → 25.9%	71.1% → 67.9% [60.1, 75.3]	20.6% (417) → 58.9% (241)
Llama-3.1-8B	73.2%	33.5% [26.8, 40.9]	40.7% [33.6, 47.4]	76.1% → 52.3%	78.5% → 67.6% [60.4, 74.5]	44.5% (317) → 21.8% (243)
Gemma-2-9B	68.0%	59.0% [51.0, 66.7]	19.9% [13.5, 26.5]	71.4% → 28.7%	74.3% → 62.8% [55.2, 70.1]	35.2% (403) → 53.8% (279)
GLM-4-9B	92.8%	50.9% [44.2, 57.6]	35.7% [29.6, 41.7]	93.1% → 37.4%	93.8% → 87.7% [82.9, 92.1]	84.3% (51) → 50.6% (87)
Yi-1.5-9B	83.4%	39.7% [31.9, 48.0]	57.9% [49.9, 66.4]	83.4% → 58.7%	85.1% → 82.5% [76.5, 88.0]	100.0% (142) → 82.4% (131)
DeepSeek-7B	66.8%	67.2% [59.9, 74.7]	29.9% [22.7, 37.0]	67.7% → 30.4%	70.9% → 36.4% [29.7, 43.5]	89.7% (252) → 45.7% (479)

Table 22: Edits that change only the column names of the S+ base tables (salient attribute optimal, $N \geq 3$, near-ID-free sources). Swap: the names of the salient column and of the lowest-EIG column are exchanged, so the salient name heads the least informative values and the partner name the best ones; a model that reads values asks the partner name. Anonymised: names replaced by `attr_i`; the salient column stays optimal. Last column: share of non-optimal draws that land on the attribute the model asks most, other than the salient one, when shown no records (Table 11), mapped back to the original name after anonymisation (number of such draws in parentheses). The 32B model was not run on these sets.

experiment	tables and selection rule	tasks run	unique tables scored	reported in
benchmark (Table 1)	292 source-capped tasks from the pool; original tasks of near-ID sources replaced by rebuilt ones; ties and identical tables removed (§3.1)	292 + 76	238	§4.1
second sample (Table 18)	near-ID-free sources, $N \geq 3$, not tied, a non-optimal attribute exists, ≤ 60 per source; rules fixed before any run	251	251	§4.1
natural tasks: S− / S+ (Table 10)	top-up from the pool, disjoint from the benchmark: near-ID-free, $N \geq 3$, not tied; S− = salient attribute not optimal, S+ = optimal, matched per source	91 / 151	70 / 71	§4.3
extreme edit	S+ tables; salient column set to the target's value; kept if the edit makes the salient attribute non-optimal and the task not tied	82	65	§4.3
mild edit (two seeds)	S+ tables; about half of the other candidates get the target's value; same keep rule	73 each	57	§4.3
reverse edit	S− tables; salient column made distinct per candidate; kept if it becomes optimal	90	69	App. A
column-name swap	S+ tables with a lower-EIG column present for every candidate; names of the salient and the lowest-EIG column exchanged	136	135	§4.3
names anonymised	all S+ tables; column names replaced by `attr_i`	151	150	§4.3
repairs, prompts, decoding	the unmodified part of the benchmark	292	171	§4.4, App.
goal stated	unmodified benchmark and second sample	292 / 251	171 / 251	§4.4

Table 23: Every task set in the paper. "Tasks run" is the number of tasks each model was called on (k=5 draws each); "unique tables scored" is after merging identical tables, removing information-tied tasks and, for paired edits, keeping only pairs for which both the base and the edited task are scored. Extreme, mild and swap/anonymised edits share the same S+ base tables, so their rows are paired with each other as well as with the base.

	benchmark	target-invariant tables	random-target expectation	rank
Qwen2.5-32B	0.036	0.036	0.037	1 / 1 / 1
Qwen2.5-14B	0.061	0.064	0.063	2 / 2 / 2
Mistral-7B	0.076	0.075	0.077	3 / 3 / 3
Qwen2.5-7B	0.079	0.080	0.082	4 / 4 / 4
Llama-3.1-8B	0.088	0.082	0.088	5 / 5 / 5
Gemma-2-9B	0.105	0.095	0.105	6 / 6 / 6
GLM-4-9B	0.129	0.132	0.132	7 / 7 / 7
Yi-1.5-9B	0.172	0.171	0.172	8 / 8 / 8
DeepSeek-7B	0.406	0.415	0.408	9 / 9 / 9

Table 24: Choice of target. Outside τ-bench the target is the first record; regret does not depend on it, but the pool and information-tie filters do. Of the 238 benchmark tables, 217 pass both filters whichever candidate is the target. Mean regret (bit) on the benchmark, on those 217 tables only, and with every table weighted by the share of its candidates under which it passes; ranks in the same order.

	benchmark	share in Movies + rebuilt Flights	without them	second sample	share in Movies	without Movies
Qwen2.5-32B	0.036	86%	0.007	0.109	91%	0.012
Qwen2.5-14B	0.061	89%	0.009	0.141	65%	0.064
Mistral-7B	0.076	62%	0.041	0.125	67%	0.054
Qwen2.5-7B	0.079	65%	0.038	0.151	43%	0.111
Llama-3.1-8B	0.088	58%	0.052	0.199	50%	0.128
Gemma-2-9B	0.105	64%	0.053	0.288	66%	0.128
GLM-4-9B	0.129	30%	0.127	0.293	29%	0.269
Yi-1.5-9B	0.172	30%	0.169	0.324	28%	0.301
DeepSeek-7B	0.406	18%	0.468	0.577	19%	0.607

Table 25: Concentration of regret by source (mean regret in bit; shares are of a model's total regret). For the two largest Qwen models most of the regret on the benchmark comes from SGD Movies and the two rebuilt SGD Flights sources; for the three weakest models it does not.

	tables	P(wrong record), regret $= 0$ [95% CI]	regret > 0.8 bit [95% CI]	questions to resolve, $0 \to > 0.8$	resolved	in the minimum rounds
Mistral-7B	87	11.7% [6.6, 17.5] (61)	41.7% [0.0, 75.0] (3)	$1.22 \to 1.67$	100.0%	66.9%
Llama-3.1-8B	87	23.8% [16.8, 30.6] (52)	55.9% [47.3, 62.2] (25)	$1.47 \to 1.92$	100.0%	36.2%
Gemma-2-9B	87	19.7% [12.5, 27.3] (44)	55.6% [47.5, 61.7] (27)	$1.39 \to 1.96$	100.0%	37.4%
GLM-4-9B	87	18.3% [12.3, 24.8] (61)	50.3% [36.1, 62.6] (15)	$1.32 \to 1.69$	90.0%	53.9%
DeepSeek-7B	87	26.0% [16.0, 36.0] (30)	64.2% [57.2, 70.0] (33)	$1.66 \to 3.20$	98.9%	34.5%

Table 26: Cost of a poor first question. Five models were run for up to $|\mathcal{A}|$ rounds on 87 tables (near-ID-free sources, a gold attribute exists) from the set of tables on which the first model studied had made a non-optimal zero-shot choice, so the set is enriched for hard tables; five trials per table. After each question the candidates are filtered on the target's true value of the asked attribute (no simulated user). P(wrong record) is the chance of committing to a wrong record if a pipeline stopped after one question and chose uniformly among the candidates still consistent, by the regret of that first question (number of tables with such a first question in parentheses; Mistral-7B has three). Unit = table, 95% task-level bootstrap.

B PROMPTS

The blocks below are generated from the code that was run. <RECORDS> and <ATTRIBUTE NAMES> are filled in per task.

Original prompt (all main results).

```
You are a customer-service agent. A customer is calling about one of
their past records (an order or a reservation) but you don't know which one yet. Here are all
their records on file (you do NOT know which one they mean):

<RECORDS: one line per record -- index, record id, JSON of its attributes>

Propose ONE natural clarifying question to ask the customer that would help you figure out which
record they mean. Pick whichever attribute you think is most likely to distinguish between these
records -- there is no single "correct" attribute, decide for yourself. Do not ask for the record's
system ID (assume the customer doesn't have it handy).
Output strict JSON: {"attribute": "<one of: <ATTRIBUTE NAMES>>", "question": "<the question to ask, in
natural language>"}.
```

Paraphrase 1 (terse, no persona).

```
Below are several records. A user is referring to exactly one of them, and you do not know which.

<RECORDS: one line per record -- index, record id, JSON of its attributes>

Ask the user ONE question about a single attribute so that their answer tells you which record it is.
Choose the attribute yourself. You may not ask for the record's ID.
Output strict JSON: {"attribute": "<one of: <ATTRIBUTE NAMES>>", "question": "<the question to ask, in
natural language>"}.
```

Paraphrase 2 (different persona, goal stated).

```
You work on a support desk. Someone has contacted you about one of the entries listed below, but has not
said which one, and you cannot tell from context.

<RECORDS: one line per record -- index, record id, JSON of its attributes>

Your aim is to identify the right entry with as few questions as possible. Decide which single attribute
to ask about first and phrase it as a natural question. The person does not have the entry's system ID to
hand, so do not ask for it.
Output strict JSON: {"attribute": "<one of: <ATTRIBUTE NAMES>>", "question": "<the question to ask, in
natural language>"}.
```

Paraphrase 3. The original wording with the records rendered as a markdown table (one row per record, one column per attribute).

Exact counts. The original prompt, with one line appended to the records; nothing else tells the model what the line is for:

```
[tool output] number of distinct values of each attribute across the <N> records above -- <attr>: <count>, ...
```

Goal stated (variant obj, Table 21).

```
You are a customer-service agent. A customer is calling about one of
their past records (an order or a reservation) but you don't know which one yet. Here are all
their records on file (you do NOT know which one they mean):

<RECORDS: one line per record -- index, record id, JSON of its attributes>

Your goal is to find out which record they mean. Ask ONE clarifying question about a single attribute.
Choose the attribute whose answer will, on average, leave the fewest records still possible, whichever
record the customer means. The customer knows the value of every attribute of their record. Do not ask
for the record's system ID (assume the customer doesn't have it handy).
Output strict JSON: {"attribute": "<one of: <ATTRIBUTE NAMES>>", "question": "<the question to ask, in
natural language>"}.
```

Instruction to count.

```
You are a customer-service agent. A customer is calling about one of
their past records (an order or a reservation) but you don't know which one yet. Here are all
their records on file (you do NOT know which one they mean):
```

```
<RECORDS: one line per record -- index, record id, JSON of its attributes>

Propose ONE natural clarifying question to ask the customer that would help you figure out which
record they mean. Do not ask for the record's system ID (assume the customer doesn't have it
handy).

Before answering, think step by step: for each candidate attribute in {<ATTRIBUTE NAMES>}, count how
many DISTINCT values it takes across all the records shown above. Prefer the attribute with the
MOST distinct values (the one most likely to separate the records), unless it is clearly not a
natural thing to ask a customer.

After your reasoning, output strict JSON on its own line: {"attribute": "<one of: <ATTRIBUTE NAMES>>",
"question": "<the question to ask, in natural language>"}.
```