Generating Questions Under Discussion with Reinforcement Learning using Ranking and Scoring for Reward and Evaluation

Kelvin Han and Claire Gardent[1]
[1]CNRS/LORIA
kelvin.han@posteo.net, claire.gardent@loria.fr

Abstract

There is growing research interest in Questions Under Discussion (QUD), a linguistic framework for representing discourse in the form of natural language question-answer pairs, which are more easily understandable and have been found useful in several applications. Our goal in this work is to improve on the quality of automatic QUD generation. As a way to sidestep the paucity of data currently, we propose a reinforcement learning-based approach using the Group Relative Policy Optimisation (GRPO) objective for LLM post-training on the task. To get there, we: (i) carefully investigated five promising methods for reference-free automatic QUD evaluation, (ii) proposed a novel prompting strategy (**SCRS**) involving ranking and scoring with structured outputs that enables QUD evaluation close to the human upperbound, (iii) leveraged findings from (i) with (ii) for the knowledge distillation from a very large LLM to obtain a more resource-efficient reward model, and which (iv) we then used in the GRPO post-training for 3B LLMs on the QUD generation task (Figure 1). Our QUD generators give overall higher-quality QUDs compared to the SOTA which is based on supervised fine-tuning; all of these are achieved using only three annotated exemplars in the few-shot prompting for evaluation, and without the use of any other annotated questions for training the QUD generators. Our code, models, and annotated examples can be found at `https://github.com/hankelvin/grpo_qud_generation`.

1 Introduction

Questions Under Discussion (QUD) (Roberts, 2012; Kuppevelt, 1995) is a linguistic framework for representing discourse structure through natural language question & answer (QA) pairs, with the view that a piece of discourse can be seen as (and hence could also be represented by) a progression through a series of information-seeking questions (typically implicitly-posed) that are raised and resolved across a piece of discourse. *We use QUD throughout this paper to refer to the framework and QUD (or the notation q) to refer to a single QUD question. A list of the QUD-specific terms used in this work, together with their notations and definitions, can be found in Table 7 in the appendices.*

Within QUD, several perspectives regarding the formalisation of how the QUD set should be constructed have been surfaced (Fu, 2025) – one of which gives rise to a dependency structure (Ko et al., 2022);[1] and which is the view we take on

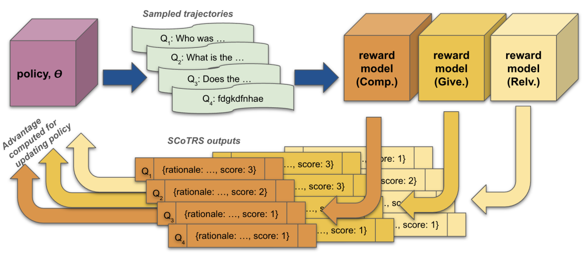

Figure 1: Overview of our approach using GRPO for post-training a QUD generation model ($\mathbf{G_{rm}}$).

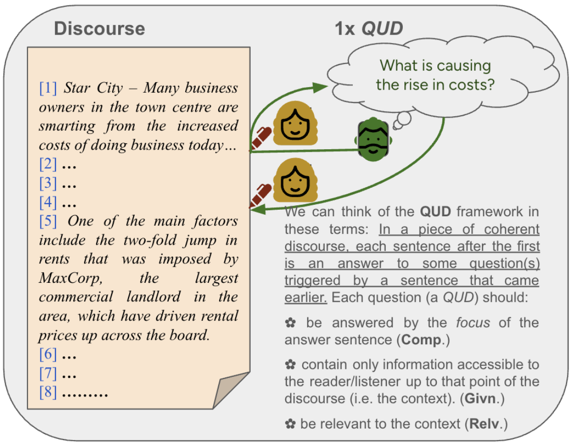

Figure 2: QUD framework and task of QUD generation.

[1]This gives a shallower structure, as opposed to say a tree structure from the view taken by (Riester, 2019), where some

Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics, pages 589–615
December 20-24, 2025 ©2025 Association for Computational Linguistics

in this work. In this view, given a piece of coherent discourse D and $s_i \in D, i > 0$ (where s_i is a sentence in the discourse), every $s_i (i \geqslant 2)$, can be seen as an **answer** (ans) to a QUD q that is triggered by $s_j (j < i)$ (which is referred to as the **anchor**, anc), and the q may contain or refer to information from anc, as well as all the sentences that came before it (which is referred to as the **context**, C). From a generation perspective, the task of automatically producing the ordered set of QUDs for a given D (such as a news article) can be broken down into two sub-tasks. The first is the identification of (anc, ans) pairs in D which can be connected by a QUD q; the second is the generation of q. We refer to the task as QUD construction, and the sub-tasks as (i) (anc, ans) identification; and (ii) QUD generation respectively.

From a computational linguistics perspective, existing discourse representation frameworks such as Rhetorical Structure Theory and Penn Discourse TreeBank (Mann and Thompson, 1988; Prasad et al., 2008) require specialised linguistic skills and training for annotation and interpretation; whereas, representing discourse through QA pairs can reduce the effort required for annotation and also facilitate inspection (and usage) beyond trained linguists. In practical terms, QUD (and QUD-style questions) has been found useful (i) as a content plan to guide text generation with fewer hallucinations and errors (Narayan et al., 2023), (ii) in analysing phenomena in text simplification such as information loss (Trienes et al., 2024) and elaborative simplification (Wu et al., 2023b), (iii) applications to aid readers' comprehension of documents (Newman et al., 2023; Cui et al., 2024), and (iv) for a metric of discourse similarity (Namuduri et al., 2025).

Nonetheless, QUD annotation remains a challenge – despite its relative annotation simplicity, it still comes with a significant cognitive load on the part of a human annotator to read and reason over the discourse.[2] As a result, there is only one large-scale annotated QUD dataset (Ko et al., 2022), and such lack of data caps further research into QUD.

One promising direction for overcoming this data constraint is in the use of reinforcement learning (RL). Recent work shows that RL post-training of LLMs helps align models to human preferences (Ziegler et al., 2020) as well as induce/unlock reasoning abilities (Wang et al., 2024). Notably, Group Relative Policy Optimisation (GRPO) (Shao et al., 2024; DeepSeek-AI et al., 2025) has been demonstrated as an efficient objective for RL post-training on coding, mathematics and logic problems (Xie et al., 2025; Lambert et al., 2025). These successes stem in part because there exists scalable verifiers to obtain reward signal (unambiguous, often binary i.e. correct/incorrect) for RL on such problems. It has not been established whether and to what extent GRPO could be applied to linguistic annotation tasks such as QUD generation, where judgements can be gradated and complete agreement between multiple annotators seldom occurs.

In this work, we study this direction of leveraging GRPO for QUD generation and make the following contributions: **[1]** ▸ We systematically evaluate approaches for automatically evaluating QUDs in a reference-free manner ("ref-free QUD evaluation"). By casting QUD evaluation as a scoring & ranking task and using a novel LLM-prompting strategy (**SCRS**, see Section 4), we show that it is possible to obtain ref-free evaluation performance near the human upperbound, which is critical as it gives us a means to obtain reward signal at scale for RL training. **[2]** ▸ We fine-tune compact (3B) LLMs for the ref-free QUD evaluation with **SCRS** by using knowledge distilled from a closed-sourced LLM (GPT4o) for a more resource-efficient reward model (RM). **[3]** ▸ We propose a GRPO post-training approach using the knowledge-distilled RM for our QUD generation model. Our approach relies on only three exemplars for **SCRS** prompting to obtain score and ranking preferences, and does not require any annotated questions for training the generator; yet it outperforms the current state-of-the-art (SOTA) based on supervised fine-tuning (SFT).

2 Related work

QUD generation and automatic evaluation Existing approaches for QUD construction (and by extension, generation) are based on supervised fine-tuning (SFT) and rely on a medium-sized corpus of

QUDs may hold relations to others in a hierarchical manner.

[2]It requires careful reading and comprehension of a piece of discourse (up to multiple paragraphs), followed by reasoning over the many pieces of information in the established ground, before crafting the QUD as a question likely to be raised at that point in the discourse, and then repeating this over the entire discourse. A well-formed QUD must also meet certain criteria, including: (i) it has to be resolvable by the focus of the ans; and (ii) its information content should mostly be from the anc triggering it. Certain strategies for optimising the criteria move in opposing directions and therefore the crafting of a QUD is an intricate balance between them.

human-annotated QUDs – the work by (Ko et al., 2023) and the SOTA by (Suvarna et al., 2024) are all trained on ***DCQA*** (Section 3.1). Although LLM-prompting has enabled strong performance on many traditional natural language processing (NLP) and generation tasks (Brown et al., 2020), and could potentially serve as a more generalisable method for QUD generation, earlier efforts by (Wu et al., 2023a) with GPT4 prompting found it to give mixed performance (giving QUDs that do well on some criteria such as answerabilty, but poorer on the others). As such, the current state-of-play on the task is one where (i) the lack of annotated data constrains efforts to automate the generation and evaluation of QUDs; and (ii) current LLMs have not been able to adequately address the needs of QUD construction and its sub-tasks. Automatic evaluation for QUDs also remains in a nascent stage, with existing work (Wu et al., 2023a; Suvarna et al., 2024) relying on methods such as rules-based lexical matching, LLM-prompting and natural language inference models that are either overly rigid or require tuning over some data to get close to human performance.

LLM-as-judge and ranking The use of LLM-as-judge (Zheng et al., 2023a) for ranking (LLM-as-judge-ranker) has been explored in information retrieval (Sun et al., 2023; Pradeep et al., 2023a,b) for (re-)ranking the relevance of documents retrieved for a query; and panels of LLM-as-judge(-rankers) to select from the generation candidates of multiple LLMs (Li et al., 2024; Han and Gardent, 2025). However, initial work here focused on judgement efficiency and have the models generate the preference ranking only without exploiting the use of test-time compute (TTC) methods (Snell et al., 2024) such as chain-of-thought (CoT) (Wei et al., 2023). Recent proposals for generative & generalist reward models (Zhang et al., 2025; Liu et al., 2025b) show that TTC enables stronger reward modeling performance; though our work (**SCRS** prompting, Section 4) extends the cases investigated (per-instance/pairwise and on verifiable tasks), and we go further in using it empirically in our RL post-training (Section 5).

Post-training with RL The prevalent methods in RL from human feedback (RLHF) – on-policy ones like Proximal Policy Optimization (PPO) (Schulman et al., 2017) and off-policy ones such as Direct Preference Optimization (DPO) (Rafailov et al., 2024), are used with human preferences of one over another in a pair of generated text. Using GRPO, and by focusing on math, coding and logic problems that can be machine-verifiable at scale (RL with verifiable rewards, RLVR), recent work (DeepSeek-AI et al., 2025; Shao et al., 2024) have been able to unlock novel reasoning capabilities to obtain meaningful performance gains on such problems. GRPO (see Section 5) is an on-policy method and requires some way to obtain reward signal on candidate generations (or "trajectories") from the LLM (the "policy"); it works optimally by going beyond pair-wise comparisons to one of sets (i.e. the "group" in its name). Since autoregressive LLM generation is stepwise and a bottleneck, being able to compute advantage at a group level also permits parallelisation to speed up training. Obtaining preference data over a set is however more complex (and might be harder to do so with humans for some tasks); one way to do so at scale could be through the use of LLMs – RL with AI feedback (RLAIF) has been demonstrated to help in safety and helpfulness alignment in dialogues (Bai et al., 2022) and NLP tasks such as summarisation (Guo et al., 2024; Lee et al., 2024).

Criteria	**Num cands**			**Total**
	2	3	4	
Comp.	93	70	38	201
Givn.	111	56	33	200
Relv.	125	63	35	223

Table 1: $\boldsymbol{QE}_{rank}$: Distribution of the number of rankable instances by number of q candidates. Number of instances varies over the criteria because we exclude instances where all its QUDs have the same score.

3 Approach

We approach the challenge of improving QUD generation by first carefully assessing a set of automatic evaluation strategies that could provide us with a scalable means to evaluate QUDs for obtaining RL training reward signal (Part I, Section 4). This includes casting ref-free QUD evaluation as a ranking task – which also lends itself well to the GRPO objective. Next, we distill the knowledge of GPT4o to obtain a ranking reward model which we use in the GRPO post-training for our QUD generation model (Part II, Section 5).

3.1 Data

We make use of three English datasets, which we describe below. In Part I we use the (C, anc, ans) and QUDs in $\boldsymbol{QE}_{rank}$ to assess ref-free QUD evaluation approaches. In Part II we use only the (C, anc, ans) of ***DCQA***'s train split in the processes to get our QUD generator, and then $\boldsymbol{QE}_{rank}$ and $\boldsymbol{TQ}_{ans}$ for evaluating our QUD generator.

[1] ▸ ***DCQA*:** (Discourse Comprehension by Question Answering (Ko et al., 2022)) is composed of 22,394 QUDs collected over 606 news articles with the help of three expert annotators alongside crowd workers on Amazon Mechanical Turk.

[2] ▸ $\boldsymbol{QE}_{rank}$**:** is drawn from QUDEval (Wu et al., 2023a), which itself comprises 2,040 human judgements of QUDs obtained from four models/approaches, including LLM-prompting with GPT4, ChatGPT and Alpaca 7B as well as an SFT-based model by (Ko et al., 2023). These QUDs were generated on the (C, ans) of a subset of the articles in the validation and test splits of ***DCQA***, and then rated on a set of criteria about their quality (see Section 4 and Table 7 for the criteria and their definitions) by a group of three linguists familiarised with the QUD annotation task. Each QUD in QUDEval was given a score of 1 (best), 2 or 3, which we reverse in this work (i.e. 3 is the best) to align with a more intuitive notion of scoring that LLMs are more likely to be familiar with. There is a set of QUDs Q for every (C, anc, ans) tuple within QUDEval,[3] and we use the part of the dataset that can form meaningful ranking instances (i.e. more than one q for the (C, anc, ans), and not all having the same score; details in Appendix B.1). We refer to this subset as $\boldsymbol{QE}_{rank}$, which has the number of ranking instances shown in Table 1.

[3] ▸ $\boldsymbol{TQ}_{ans}$**:** is drawn from TEDQ (Westera et al., 2020), which itself contains crowdsourced questions on the transcripts for six talks given in the TED talks series. It was collected as the set of questions evoked as the speech progresses; a portion of them are QUDs because they were able to be matched to a sentence later in the speech as its answer. TEDQ was leveraged by (Wu et al., 2024) in a QUD-related dataset with additional annotations of question salience and answerability; although there they only used a single transcript. We use the part of TEDQ where the (anc, ans) pairs were identified to have answerable questions, and refer to this as $\boldsymbol{TQ}_{ans}$ (details in Appendix B.2).

[3]Each $|Q|$ varies from 1 to 4 (see Appendix B.1).

4 Part I: Ref-free QUD evaluation

QUD criteria Following (Wu et al., 2023a), we focus on three criteria for assessing the quality of a given QUD, namely: **(i)** Answer Compatibility (*Comp.*) – whether q is answered by the focus of ans, **(ii)** Givenness (*Givn.*) – whether q only contains information in/salient to $C+anc$, and **(iii)** Anchor Relevance (*Relv.*) – whether q mostly contains information in anc; see Table 7 for their definitions.[4] Note that the human annotations in QUDEval allowed (Wu et al., 2023a) to compute pairwise macro-averaged F1 between the annotators' scores, which gives the human upperbounds on the task of QUD evaluation (0.60-0.61 for each criterion).

Reference free QUD evaluation approaches Our goal here is to identify a reference-free method that can be used to obtain the reward signal needed for GRPO post-training. The five broad classes of approaches we examine include two used in previous work – **(1)** `RB-NLI`, a set of rules-based & natural language inference (NLI) methods (Suvarna et al., 2024) and **(2)** `JDG`, an LLM-as-judge approach proposed by (Wu et al., 2023a) that judges one q at a time (i.e. singly). In addition, we propose a variant $\texttt{JDG}_{fs}$ that matches the 3-shot prompting available for two alternative approaches, namely: **(3)** `RANK` (or LLM-as-judge-ranker) which uses LLM-as-judge but casts QUD evaluation as a ranking task on a set, and **(4)** `RANK-PANEL` which is LLM-as-judge-ranker using a panel of compact LLMs of ⩽14B[5] (Han and Gardent, 2025). Finally, we have **(5)** `LLMQA (logp)` that uses the log probabilities of an LLM to assess *Comp.*. Further details on these five approaches are in Appendix C.

SCRS: Obtaining scores and rationales from LLM-as-judge-ranker We extend LLM-as-judge-ranker by prompting the LLM to first produce a rationale and a score for each of the q candidates, and then give the ranking. This can be seen as a form of CoT and is aligned with recent findings that test-time compute (TTC) can improve LLM performance on reasoning tasks (Wu et al., 2025; Snell et al., 2024). We also prompt the LLM to do

[4]We take the same view as (Wu et al., 2023a) and set aside the fourth criterion – Language Quality (`Lang.`) – as LLM-based QUD generation approaches show the ability to generate fluent questions that meet this criterion.

[5]The panel consists of these models: `microsoft/phi-4`, `meta-llama/Llama-3.1-8B-Instruct`, `google/gemma-2-9b-it` and `Qwen/Qwen2.5-7B-Instruct` on `https://huggingface.co/`

	(1) RB-NLI	(2a) JDG	(2b) $\texttt{JDG}_{fs}$	(3) RANK	(4) RANK-PANEL	(5) LLMQA (logp)
n-shot	-	-	3	3	3	2
n-shot CoT/**SCRS**	-	-	3	3	3	-

Table 2: The prompting strategies explored for each class of reference-free QUD evaluation approach. "n-shot" and numbers in cells refer to the number of exemplars used in the prompting. "CoT" denotes Chain-of-Thought; for a description of **SCRS** see Section 4. *We use n-shot = 3 where possible; except for* LLMQA (logp) *in the first row, which is set at 2-shot due to resource constraints. In the second row, the* RANK *and* RANK-PANEL *methods use **SCRS**, whereas the* $\texttt{JDG}_{fs}$ *method there only uses CoT and ranks each QUD singly, i.e. it provides ablation to the SCRS prompting.*

so in a structured manner that facilitates extracting each piece of information. We call this approach Structured CoT for Ranking & Scoring (**SCRS**); examples of the prompting for **SCRS** (Appendix F) and its output (Table 12) are in the appendices.

4.1 Metrics

(i) Macro F1 & (ii) Score-Rank Consistency We follow existing work (Wu et al., 2023a; Suvarna et al., 2024) and use macro F1 to validate these ref-free QUD evaluation measures, as it indicates how well a measure can predict human scores accurately. Since LLM generations could include errors, one concern was whether, with the **SCRS** setting, the scores provided and the eventual ranking by LLMs are consistent – hence, we use a measure we refer to as Score-Rank Consistency to assess this.[6]

Other measures For a fuller understanding of each approach's performance – especially regarding their applicability for use in GRPO post-training – we also compute **(iii) Net Discounted Cumulative Gain (NDCG)** and **(iv) Top-Rank Accuracy**, which respectively informs us about how good an approach is at: (i) ranking members of a set relative to each other, and (ii) identifying the best/one of the best within the set. These metrics are further explained in Appendix C.2.

[6]Since the scores fall in range of 1,2 and 3, there can be ties in the scores amongst the sets of (up to 4) QUDs being ranked, we checked if the returned ranking is in the set of all possible ranking orders given the scores generated.

4.2 Assessing the evaluation approaches

We explored up to two prompting strategies for each of the five approaches which is summarised in Table 2. Note that, for space, we focus our discussion hereon on the comparison between JDG/$\texttt{JDG}_{fs}$[7] and RANK (with Llama 8B, Qwen 7B, GPT4o and Claude Opus 4 variants). This is because we found RANK to perform the best overall and use it in Part II; their macro F1 and Score-Rank Accuracy results can be found in Table 3 (top rows); detailed results for all five classes of approaches can be found in Tables 10 & 11 in the appendices. Notably, we found the following:

[1] ▸ without learning a mapping function over some validation data, JDG (GPT4o) of (Wu et al., 2023a) is far from the human upperbound of ∼0.6 macro F1 (see Table 4.1).

[2] ▸ the various ways of using GPT4o clearly outperforms RB-NLI, RANK with the 7/8B LLMs (Qwen and Llama) and even RANK-PANEL (see also Tables 10 & 11).

[3] ▸ RANK-PANEL does not give consistently better results over RANK using a 7/8B LLM (e.g. compare RANK with Qwen 7B, Tables 10 & 11); this may be due to the complexity of the ranking task here.

[4] ▸ prompting with CoT/**SCRS** generally gives better ranking performance (NDCG and Top-Rank Accuracy), although it could lead to a dip in ranking performance for *Comp.* which assesses whether a QUD is fully/partially/not answered by its ans.

[5] ▸ Using RANK with GPT4o and **SCRS** (RANK (GPT4o) hereon) gives macro F1 close to the human upperbound for *Comp.* and *Givn.*, and substantially higher for *Relv.*. Furthermore, RANK (Claude Opus 4) (i.e. using Claude Opus 4 instead) goes on to lift performance on *Relv.* to reach close to the human upperbound, and on *Givn.* to even exceed the human upperbound. Both RANK (GPT4o) and RANK (Claude Opus 4) also also give perfectly consistent score-rank responses (see Table 4.1). These gains with **SCRS** could be due to (i) the task-set up facilitating more globally consistent scoring (via useful context from the multiple QUDs presented, allowing the LLM to score them relative to others instead of on their own), and (ii) additional TTC gains from the longer generations.

[7]Following (Wu et al., 2023a), except that they used GPT4 and we updated to use GPT4o.

		Comp.		Givn.		Relv.	
		F1	SRC	F1	SRC	F1	SRC
Off-the-shelf	JDG (G)	0.16	-	0.26	-	0.29	-
	JDG$_{fs}$ (G)	0.54	-	0.31	-	0.42	-
	RANK (L)	0.44	0.93	0.39	0.96	0.33	0.92
	RANK (Q)	x	x	x	x	x	x
	RANK (G)	0.59	1.00	0.59	1.00	0.47	1.00
	RANK (C)	**0.59**	1.00	**0.75**	1.00	**0.58**	1.00
K/Distil.	GRPO (Q)	0.47	1.00	0.36	1.00	0.35	1.00
	SFT (L)	x	x	x	x	x	x
	SFT (Q)	**0.57**	1.00	**0.57**	1.00	**0.43**	1.00

Table 3: Macro F1 (F1) and Score-Rank Consistency (SRC). (G), (C), (L) and (Q) denote GPT4o, Claude Opus 4, Llama 8B and Qwen 7B respectively. x denotes significant numbers of instances with **SCRS** outputs where parsing fails. Top rows are off-the-shelf use of these LLMs, and bottom rows are of the knowledge distillation models of Section 5.

5 Part II: GRPO for QUD generation

The approach for our QUD generation model comprises two steps. The first step involves creating **SCRS** outputs from a very large LLM to then obtain a compact-sized reward model (RM) via knowledge distillation (KD); the intent of this step is for a way to sidestep resource-heavy calls to GPT4o to obtain reward signal. In the second step, we use the GRPO training objective together with the compact RM to post-train 3B LLMs for QUD generation.

Knowledge distillation for an RM This first step involves two parts. Firstly, we used RANK (GPT4o) to obtain KD data for the ref-free QUD evaluation task.[8] Briefly (with details in Appendix D.1), we ran an initial round of GRPO post-training on the Llama 3B model using (C, anc, ans) tuples from the ***DCQA*** train split. From sending the q of the sampled trajectories to RANK (GPT4o) for evaluation, we could collect **SCRS** outputs usable for KD. We explored two approaches for KD with these outputs: (i) SFT using sequence-level KD (Kim and Rush, 2016) on 3B Llama and Qwen models, as well as (ii) GRPO post-training on Qwen 3B. At a resource-equivalence level (GPU compute) we find that SFT with Qwen 3B (RM$^{Qwen}_{SFT}$) is best, and gives performance close to RANK (GPT4o) (Table 3, bottom rows).

GRPO post-training In this second step, we post-train 3B-sized LLMs (Qwen & Llama)[9] to produce a reasoning underlying the generation of the QUD, before giving the $\hat{q}$. The training process involves exploration through prompting the LLM for candidate generations (i.e. underlying reasoning and $\hat{q}$, or "trajectories") using sampling with temperature, and optimising with the GRPO objective, which includes maximising the rewards obtained by the trajectories in reward functions we set (including for *Comp.*, *Givn.* and *Relv.* scored with RM$^{Qwen}_{SFT}$). More completely, the GRPO objective is of this form:

$$-\frac{1}{\sum_{i=1}^{G}|o_i|}\sum_{i=1}^{G}\sum_{t=1}^{|o_i|}\left[\frac{\pi_\theta(o_{i,t}|x,o_{i,<t})}{[\pi_\theta(o_{i,t}|x,o_{i,<t})]_{\text{no grad}}}\hat{A}_{i,t}-\beta D_{\text{KL}}[\pi_\theta\|\pi_{\text{ref}}]\right] \quad (1)$$

where x is the prompt that also contains the (C, anc, ans) we want to generate a $\hat{q}$ for, G is the set of sampled trajectories from the LLM (or "policy" π_θ) for x in one training step, $\hat{A}$ is the relative advantage computed over the rewards obtained for the trajectories, and the D_{KL} term is the KL divergence between the per-token log probabilities of G of the current (π_θ) and the reference policy (π_{ref})[10] which is intended to keep the updates to the LLM within a "trusted" zone (that ideally can already emit valid, or reasonably close to valid, trajectories). The post-training for each LLM took about 40 hours, and was done with one H100 set up with asynchronous communication with three RTX 4500 (RM$^{Qwen}_{SFT}$ for each criterion). Figure 1 gives an overview of this step; with information on the reward functions, hyperparameters and implementation details (including a cold-start stage) for the GRPO post-training of this part in Appendix D.2.

5.1 Experimental set-up

Baselines/Ablations We compare our approach (**G$_{rm}$**, short for GRPO with RM$^{Qwen}_{SFT}$) against QUD-Select (**QS**) (Suvarna et al., 2024), which is SOTA on the task of QUD construction. **QS** uses the (C, anc, ans, q) tuples from ***DCQA*** for fine-tuning ~8B LLMs to (i) identify $\hat{anc}$ given (C, ans), and (ii) generate $\hat{q}$ given $(C, \hat{anc}, ans)$. We used their code[11] to obtain 7B/8B Llama and Qwen versions, as well as 3B ones comparable to the 3B models we post-train with GRPO. We also compare **G$_{rm}$** against (i) **ZS**, which is using zeroshot LLM-prompting and (ii) **G$_{rb}$**, which is GRPO post-training using the RB-NLI method for reward modeling throughout. These two also serve as ablations

[8] We used GPT4o throughout the data generation and post-training in Part II, as these were before the release of Claude Opus 4 (which remains significantly more costly to access, with API charges at least 7.5 times higher than GPT4o).

[9] meta-llama/Llama-3.2-3B-Instruct and Qwen/Qwen2.5-3B-Instruct on https://huggingface.co

[10] In our case, is set to the base model throughout training.

[11] https://github.com/asuvarna31/qudselect

on the use of GRPO post-training and RM_{SFT}^{Qwen}.

Evaluation data We evaluate **$\mathbf{G_{rm}}$** and the baselines by having them generate QUDs over the set of (C, anc, ans) in $\boldsymbol{QE}_{rank}$, as well as in $\boldsymbol{TQ}_{ans}$ (to evaluate out-of-distribution performance). Given how the QUDs in QUDEval were produced (Appendix B.1) it is possible that not all (C, anc, ans) in $\boldsymbol{QE}_{rank}$ licenses a perfect QUD, therefore we also investigated the approaches' performance on the subset of (C, anc, ans) tuples having a human-written QUD collected ($\boldsymbol{QE}_{rank}^{human}$).

Evaluation measures We use two measures – RANK (GPT4o) and `Classifier`, on the outputs from the approaches within each model family. `Classifier` is a set of models fine-tuned on QUDEval, that was proposed by (Suvarna et al., 2024) for scoring each of the criteria; we use it here similar to the way multiple NLG metrics such as BLEU, ROUGE etc are used to obtain signals to assess approaches being compared.[12] Since we have 3B and 7B/8B variants of the **QS** models, we ran the comparisons twice – this gives us: (i) scores of each approach within each model family, as well as (ii) further insights about RANK (GPT4o) stability. Subsequently, we used RANK (Claude Opus 4) on the outputs of the Qwen set of models for further verification of **$\mathbf{G_{rm}}$**'s performance against the **QS** baseline.[13]

Human evaluation We also carried out a human evaluation on the outputs of Qwen and Llama variants – the **QS** baseline (7B/8B versions) and our **$\mathbf{G_{rm}}$** models (3B). We did this for 47 (anc, ans) tuples (i.e. for a total of 188 QUDs) that are spread over 8 articles. One of the authors, who is familiar with the QUD annotation framework, and who had produced the exemplars used for **SCRS** prompting – scored the models' generated QUD for each criteria using the annotation platform released by (Wu et al., 2023a). The QUDs were grouped by (C, anc, ans) to ease annotation load, but the order of the systems were randomised differently for each group (i.e. the annotator was blind to the identity of the models). We then followed the convention above (Section 3.1) and reversed the direction of the scores so that higher values are better.

5.2 Results and analysis

The RANK (GPT4o) and `Classifier` results for $\boldsymbol{QE}_{rank}$ and $\boldsymbol{TQ}_{ans}$ are in Tables 4 & 13; for each $\hat{q}$, we also obtain a global score by summing its *Givn.*, *Comp.* and *Relv.* scores, and report the mean of these (`Full`) there too. The distribution of the `Full` scores for each approach is in Figure 3; whereas the distribution of scores for each of the criteria can be found in Figures 4 & 5 in the appendices. Example outputs from the Qwen versions of **$\mathbf{G_{rm}}$** and **QS** can be found in Tables 15 & 16 in the appendices; they also have examples of the RANK (GPT4o) scores and rationales obtained. Some of the baseline models like **$\mathbf{G_{rb}}$**(Llama) generate ill-formed outputs (e.g. gibberish) and a proper question could not be extracted;[14] the "fail %" row in Table 4 indicates the percentage of such outputs. The results of the further validation with RANK (Claude Opus 4) are in Table 5, and those of the human evaluation are in Table 6.

[1] ▸ Better QUDs with GRPO post-training: The 3B-sized **$\mathbf{G_{rm}}$** outperforms the SOTA **QS** (both 3B and 7B/8B versions) to give QUDs for $\boldsymbol{QE}_{rank}$ with scores that are: (i) higher on a holistic basis (i.e. `Full` in Table 4; with statistical significance on both RANK (GPT4o) and `Classifier`), and (ii) more balanced across the criteria. Although **$\mathbf{G_{rm}}$** underperforms **QS** on one criteria (*Givn.*), it is better on the other two, including *Comp.*, which is important for a QUD (i.e. be answerable). This trend is consistent over both Qwen and Llama. It is also repeated in the `Full` scores for $\boldsymbol{QE}_{rank}^{human}$ (i.e. the human QUD-licensed subset of (C, anc, ans)) (Table 14); there **$\mathbf{G_{rm}}$** also outperforms **QS** (with significance) by 0.47-1.56 points (max score: 9) on both RANK (GPT4o) (Qwen 7B: 6.76 vs 5.38, Llama 8B: 6.52 vs 5.64) and `Classifier` (Qwen 7B: 8.05 vs 7.20, Llama 8B: 7.77 vs 7.30). Note that the SOTA performance of **QS** stems from its ability to vary the $\hat{anc}$ predicted, as a way to identify $(\hat{anc}, ans)$ that returns a $\hat{q}$ as optimal as possible with respect to the distribution learnable by the

[12]On the original $\boldsymbol{QE}_{rank}$ (which `Classifier` training data overlaps), the `Classifier` obtained macro F1 of 0.76, 0.45 and 0.68 for *Comp.*, *Givn.* and *Relv.*, supporting their use as a secondary metric here. We use the recipe and the data released by (Suvarna et al., 2024), details in Appendix D.3.

[13]Given the high costs of Claude Opus 4, we focused here on the Qwen models and only compare **$\mathbf{G_{rm}}$** against the larger 7B version of the **QS** baseline.

[14]We wrote regular expressions to extract a possible question as far as possible; failing which, we replaced the entire output with: "QUD GENERATION FAILURE (NOT EXTRACTABLE FROM OUTPUT)." before scoring with RANK (GPT4o) and `Classifier`, which aligns with the setting investigated in Part I (all items ranked were well-formed questions).

	Qwen							
Appr.	**QS**	**ZS**	$\mathbf{G_{rb}}$	$\mathbf{G_{rm}}$	**QS**	**ZS**	$\mathbf{G_{rb}}$	$\mathbf{G_{rm}}$
Size	3B	3B	3B	3B	7B	3B	3B	3B
	RANK (GPT4o)							
Comp.	1.17	1.97	2.27	**2.34**	1.19	1.96	2.29	**2.33**
Givn.	**2.56**	1.92	2.12	2.33	**2.44**	1.97	2.22	2.33
Relv.	1.56	1.75	1.78	**2.02**	1.67	1.75	1.79	**2.01**
`Full`	5.29	5.64	6.17	**6.70**	5.30	5.68	6.30	**6.66**
	`Classifier`							
Comp.	1.66	**2.76**	2.68	2.69	1.67	←	←	←
Givn.	**2.88**	2.66	2.71	2.72	**2.92**	←	←	←
Relv.	2.32	2.41	2.49	**2.54**	2.52	←	←	←
`Full`	6.86	7.84	7.89	**7.94**	7.11	←	←	←
min.	4	3	4	4	2	←	←	←
mean	10.3	25.4	12.3	13.7	7.6	←	←	←
max.	58	109	28	38	16	←	←	←
Fail	5	0	0	0	1	←	←	←
Same	52	0	0	0	46	←	←	←

	Llama							
Appr.	**QS**	**ZS**	$\mathbf{G_{rb}}$	$\mathbf{G_{rm}}$	**QS**	**ZS**	$\mathbf{G_{rb}}$	$\mathbf{G_{rm}}$
Size	3B	3B	3B	3B	8B	3B	3B	3B
	RANK (GPT4o)							
Comp.	1.20	1.96	1.00	**2.35**	1.21	1.92	1.00	**2.37**
Givn.	**2.66**	2.08	1.00	2.21	**2.71**	1.99	1.00	2.23
Relv.	1.80	1.88	1.00	**1.98**	1.77	1.89	1.00	**2.01**
`Full`	5.66	5.93	3.00	**6.55**	5.69	5.80	3.00	**6.61**
	`Classifier`							
Comp.	1.74	**2.66**	1.22	2.63	1.78	←	←	←
Givn.	**2.92**	2.66	2.25	2.65	**2.94**	←	←	←
Relv.	2.47	2.44	1.52	**2.50**	**2.50**	←	←	←
`Full`	7.12	7.76	4.99	**7.79**	7.22	←	←	←
min.	5	2	1	5	4	←	←	←
mean	8.7	24.7	10.1	17.1	9.1	←	←	←
max.	8	174	41	379	28	←	←	←
Fail	0	1	81	3	0	←	←	←
Same	44	0	10	1	61	←	←	←

Table 4: Scores for QUDs generated on (C, anc, ans) tuples in $\boldsymbol{QE}_{rank}$. ← denotes that the same values as those in the corresponding set of columns on the left applies (e.g. `Classifier` can score QUDs one at a time, hence its scores for **ZS**, $\mathbf{G_{rb}}$ and $\mathbf{G_{rm}}$ are the same). **Significance tests:** underlined denote statistical significance (p-value < 0.05 for one-tailed t-test with bootstrap re-sampling and Wilcoxon signed-test) that the approach's scores are better than **QS**.

QS model.[15] This means that despite being given some fixed (C, anc, ans) assessed to be able to hold a valid QUD (e.g. $\boldsymbol{QE}^{human}_{rank}$), the **QS** model may give QUDs far from desired (see Table 14 in the appendices). While the **QS** approach may provide QUD sets suitable for certain applications, we might want separate modules for (anc, ans) identification and QUD generation, for more stable representations.

[2] ▸ Differences in QUDs generated: **QS** has large numbers of the same $\hat{q}$ for different (C, anc, ans) tuples (up to 61% vs max 1% by $\mathbf{G_{rm}}$, see "same %" in Table 4), and its $\hat{q}$ are overall shorter (average of $\leqslant$ 10 words vs 13-17 words for $\mathbf{G_{rm}}$). These indicate its QUDs are less specific than $\mathbf{G_{rm}}$'s, which likely enable its higher *Givn.* scores albeit at the expense of its *Comp.* scores which are substantially lower than $\mathbf{G_{rm}}$'s and just above the minimum score in both $\boldsymbol{QE}_{rank}$ and $\boldsymbol{QE}^{human}_{rank}$(i.e. less specific QUDs have less room for *Givn.* errors, but may also be less targeted towards being well-answered by ans).

[3] ▸ OOD application: For $\boldsymbol{TQ}_{ans}$, only the Qwen version of $\mathbf{G_{rm}}$ score higher than its **QS** counterpart (Table 13; in `Full` on both RANK (GPT4o) and `Classifier`; with significance in the latter only). This could be due to the contents of the discourse in $\boldsymbol{TQ}_{ans}$, and characteristics of the Llama model (see next point). Compared to the news articles of $\boldsymbol{QE}_{rank}$ which are of some event and filled with details about it, TED talks are on some topic and usually covers general concepts related to the topic. In the former case, to be well-formed, QUDs often need the proper specificity of these events and their details, whereas in the latter, more general questions like those of **QS**'s (i.e. shorter and with more questions that are the same – 16% vs 0%) might suffice. Nonetheless, it is worth noting that **QS**(Qwen) has a failure rate of 9% here, compared to 0% for our $\mathbf{G_{rm}}$(Qwen).

[4] ▸ Llama vs Qwen – Base model differences: Robust GRPO post-training for QUD generation appears to depend on model family differences. $\mathbf{G_{rm}}$(Llama) has nearly four times (Figure 3) the minimum-score QUDs compared to $\mathbf{G_{rm}}$(Qwen),[16] and we also found $\mathbf{G_{rm}}$(Llama) to generate generic and simpler reasoning tokens, unlike the varied and input-specific ones by $\mathbf{G_{rm}}$(Qwen) (see Appendix N). Our findings are consistent with those of (Gandhi et al., 2025) that the Qwen 3B model, compared to Llama 3B, possesses cognitive "behaviours" that predispose it to stronger performance on math reasoning tasks, even if both models are put through the same RL post-training procedure. This could also explain the poorer OOD performance by $\mathbf{G_{rm}}$(Llama) on $\boldsymbol{TQ}_{ans}$ above.

[5] ▸ Quality of reward signal: The performance of $\mathbf{G_{rb}}$ trails $\mathbf{G_{rm}}$. Notably, in the case

[15]Their approach is based on over-generation – at inference, **QS** uses beam search to return multiple $\hat{anc}$ candidates, which are then used to generate multiple $\hat{q}$ candidates. Then the `RB-NLI` method (Section 4) is used to score the $\hat{q}$ candidates for *Comp.*, *Givn.* and *Relv.* to pick the highest-scoring one.

[16]Although $\mathbf{G_{rm}}$(Llama) does also produces slightly more max-scoring QUDs than $\mathbf{G_{rm}}$(Qwen) (about 1.25 times). One possible approach for mediating these variabilities could be to have GRPO post-training on several compact models from multiple families for QUD generation and then selecting the best QUD to return with RANK (GPT4o) or similar.

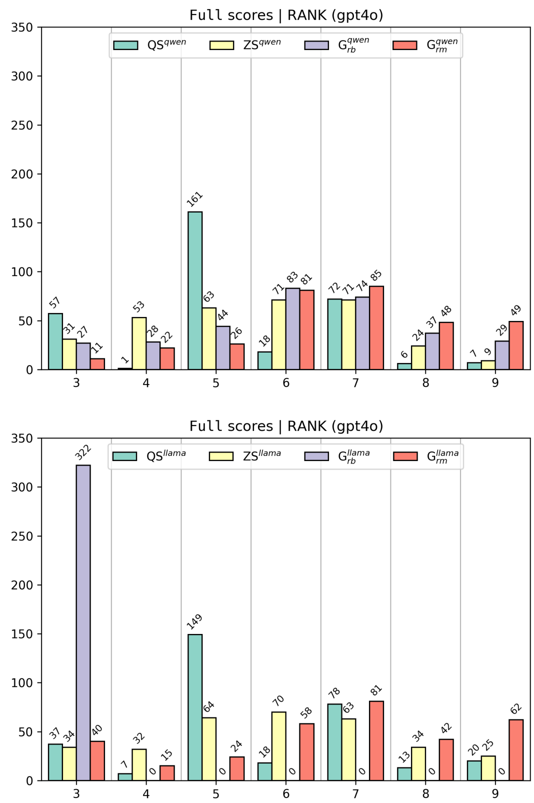

Figure 3: Distribution of global (*Comp.*+*Givn.*+*Relv.*) scores obtained with RANK (GPT4o).

of $\mathbf{G_{rb}}$(Llama), the post-training fails with completely degenerate outputs using the less accurate `RB-NLI` for reward signal; this failure however does not happen for $\mathbf{G_{rb}}$(Qwen), likely due to the model differences discussed above. Separately, we note a limitation of `Classifier`-like approaches that rely on tuning a model over some data – when used in an out-of-distribution setting with the marked-degenerate outputs of $\mathbf{G_{rb}}$(Llama) (see Footnote 12), they do not always assign the lowest score, rendering them less suitable for use in reward modeling.

[6] ▸ Further validation with RANK (Claude Opus 4) and human evaluation: The RANK (Claude Opus 4) results (Table 5) and the human evaluation (Table 6) reflect the same trend as the automatic evaluation above (see Table 4; i.e. $\mathbf{G_{rm}}$ outperforms **QS** overall (Full), and for all criteria except for *Givn.*). Both of these results further confirm (i) the use of **SCRS** prompting for reference-free QUD evaluation; and (ii) the significantly stronger performance of our $\mathbf{G_{rm}}$over the **QS** baseline.

	Qwen							
	QE_{rank}				QE^{human}_{rank}			
Appr.	**QS**	**ZS**	$\mathbf{G_{rb}}$	$\mathbf{G_{rm}}$	**QS**	**ZS**	$\mathbf{G_{rb}}$	$\mathbf{G_{rm}}$
Size	7B	3B	3B	3B	7B	3B	3B	3B
	RANK (Claude Opus 4)							
Comp.	1.20	1.75	2.16	**2.21**	1.28	1.74	**2.24**	2.17
Givn.	**2.78**	2.11	2.38	2.56	**2.76**	2.13	2.35	2.61
Relv.	1.79	1.72	1.93	**2.12**	1.77	1.73	1.92	**2.24**
Full	5.77	5.58	6.47	**6.88**	5.81	5.60	6.51	**7.02**

Table 5: Scores (obtained with RANK (Claude Opus 4)) for QUDs generated on (C, anc, ans) tuples in $\boldsymbol{QE_{rank}}$. **Significance tests:** underlined denote statistical significance (p-value < 0.05 for one-tailed t-test with bootstrap resampling and Wilcoxon signed-test) that the approach's scores are better than **QS**.

	Qwen		Llama	
	QS	$\mathbf{G_{rm}}$	**QS**	$\mathbf{G_{rm}}$
Size	7B	3B	8B	3B
Comp.	1.32	**2.57**	1.23	**2.17**
Givn.	**2.64**	2.60	**2.77**	2.19
Relv.	1.57	**2.36**	1.51	**2.09**
Full	5.53	**7.53**	5.51	**6.45**

Table 6: Results of randomised human evaluation on the quality of the generated QUDs.

6 Conclusion

We propose an RL-based approach using GRPO to obtain QUD generators based on 3B-sized LLMs that meaningfully outperform the SOTA approach using supervised fine-tuning over a corpus of ∼22k QUDs. We achieved this without using human-annotated questions for training and with only three exemplars to use in prompting for evaluation. To do this, we systematically studied approaches for reference-free QUD evaluation and proposed a novel prompting strategy for LLM-as-judge-ranker to enable automatic QUD evaluation close to the level of the human upperbound. We also used knowledge distillation from a very large LLM to a 3B model for a resource-efficient LLM-as-judge-ranker reward model in our GRPO post-training. Besides meaningful improvements to QUD generation, our work also indicates the applicability of GRPO post-training beyond verifiable problems to those with elements of variability in judgements.

7 Acknowledgments

This work received government funding managed by the French National Research Agency under France 2030, reference number "ANR-23-IACL-0004." (AI Chair: "Semantically Consistent LLM Based Text Generation"). Experiments presented in this paper were carried out using the Grid'5000

testbed, supported by a scientific interest group hosted by Inria and including CNRS, RENATER and several Universities as well as other organizations (see `https://www.grid5000.fr`).

8 Limitations

While we show that it is already possible (with $\mathbf{G_{rm}}$) to exceed SOTA QUD generation using a knowledge-distilled reward model (RM) – namely, RM_{SFT}^{Qwen} (Section 5), gains on the QUD generation task could still be possible with stronger RMs. Firstly, while having RM_{SFT}^{Qwen} facilitated our experimentation (and, which our results demonstrates, can serve as a recipe for more resource efficient reward modeling), it is not likely to be necessary for obtaining similar performance for the task (e.g. through direct use of RANK (GPT4o) or similar). Secondly, and at a more general level: an open challenge for using LLM-as-judge is the issue of position bias in LLMs in general, where the LLM's prediction may vary over the order of the items to be judged are presented to it; although we took steps to mitigate this by randomly shuffling the order of each set of candidates presented, doing so does not fully address the issue. Since we obtain the GRPO post-training reward signal through LLM-as-judge-ranker, such position bias could serve as a limit on the quality of the judgements. We do expect that, as ways to address the position bias issue in LLMs are found, it is likely that stronger performances on our approach – the reward modeling and GRPO post-training for QUD generation – might be obtainable.

References

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, and 32 others. 2022. Constitutional ai: Harmlessness from ai feedback. Preprint, arXiv:2212.08073.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. Preprint, arXiv:2004.05150.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. Preprint, arXiv:2005.14165.

Peng Cui, Vilém Zouhar, Xiaoyu Zhang, and Mrinmaya Sachan. 2024. How to engage your readers? generating guiding questions to promote active reading. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 11749–11765, Bangkok, Thailand. Association for Computational Linguistics.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 181 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. Preprint, arXiv:2501.12948.

Yingxue Fu. 2025. A survey of qud models for discourse processing. Preprint, arXiv:2502.15573.

Kanishk Gandhi, Ayush Chakravarthy, Anikait Singh, Nathan Lile, and Noah D. Goodman. 2025. Cognitive behaviors that enable self-improving reasoners, or, four habits of highly effective stars. Preprint, arXiv:2503.01307.

Shangmin Guo, Biao Zhang, Tianlin Liu, Tianqi Liu, Misha Khalman, Felipe Llinares, Alexandre Rame, Thomas Mesnard, Yao Zhao, Bilal Piot, Johan Ferret, and Mathieu Blondel. 2024. Direct language model alignment from online ai feedback. Preprint, arXiv:2402.04792.

Kelvin Han and Claire Gardent. 2025. Generating complex question decompositions in the face of distribution shifts. In Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 1189–1211, Albuquerque, New Mexico. Association for Computational Linguistics.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In International Conference on Learning Representations.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 1317–1327, Austin, Texas. Association for Computational Linguistics.

Wei-Jen Ko, Cutter Dalton, Mark Simmons, Eliza Fisher, Greg Durrett, and Junyi Jessy Li. 2022. Discourse comprehension: A question answering framework to represent sentence connections. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 11752–11764, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Wei-Jen Ko, Yating Wu, Cutter Dalton, Dananjay Srinivas, Greg Durrett, and Junyi Jessy Li. 2023. Discourse analysis via questions and answers: Parsing dependency structures of questions under discussion. In Findings of the Association for Computational Linguistics: ACL 2023, pages 11181–11195, Toronto, Canada. Association for Computational Linguistics.

Jan Van Kuppevelt. 1995. Discourse structure, topicality and questioning. Journal of Linguistics, 31(1):109–147.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V. Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Chris Wilhelm, Luca Soldaini, and 4 others. 2025. Tulu 3: Pushing frontiers in open language model post-training. Preprint, arXiv:2411.15124.

Harrison Lee, Samrat Phatale, Hassan Mansoor, Thomas Mesnard, Johan Ferret, Kellie Ren Lu, Colton Bishop, Ethan Hall, Victor Carbune, Abhinav Rastogi, and Sushant Prakash. 2024. RLAIF vs. RLHF: Scaling reinforcement learning from human feedback with AI feedback. In Proceedings of the 41st International Conference on Machine Learning, volume 235 of Proceedings of Machine Learning Research, pages 26874–26901. PMLR.

Ruosen Li, Teerth Patel, and Xinya Du. 2024. PRD: Peer rank and discussion improve large language model based evaluations. Transactions on Machine Learning Research.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. Preprint, arXiv:1907.11692.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. 2025a. Understanding r1-zero-like training: A critical perspective. Preprint, arXiv:2503.20783.

Zijun Liu, Peiyi Wang, Runxin Xu, Shirong Ma, Chong Ruan, Peng Li, Yang Liu, and Yu Wu. 2025b. Inference-time scaling for generalist reward modeling. Preprint, arXiv:2504.02495.

William C. Mann and Sandra A. Thompson. 1988. Rhetorical structure theory: Toward a functional theory of text organization. Text - Interdisciplinary Journal for the Study of Discourse, 8(3):243–281.

Ramya Namuduri, Yating Wu, Anshun Asher Zheng, Manya Wadhwa, Greg Durrett, and Junyi Jessy Li. 2025. QUDsim: Quantifying discourse similarities in LLM-generated text. In Second Conference on Language Modeling.

Shashi Narayan, Joshua Maynez, Reinald Kim Amplayo, Kuzman Ganchev, Annie Louis, Fantine Huot, Anders Sandholm, Dipanjan Das, and Mirella Lapata. 2023. Conditional generation with a question-answering blueprint. Transactions of the Association for Computational Linguistics, 11:974–996.

Benjamin Newman, Luca Soldaini, Raymond Fok, Arman Cohan, and Kyle Lo. 2023. A question answering framework for decontextualizing user-facing snippets from scientific documents. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pages 3194–3212, Singapore. Association for Computational Linguistics.

Ronak Pradeep, Sahel Sharifymoghaddam, and Jimmy Lin. 2023a. RankVicuna: Zero-shot listwise document reranking with open-source large language models. arXiv:2309.15088.

Ronak Pradeep, Sahel Sharifymoghaddam, and Jimmy Lin. 2023b. Rankzephyr: Effective and robust zero-shot listwise reranking is a breeze! Preprint, arXiv:2312.02724.

Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind Joshi, and Bonnie Webber. 2008. The Penn Discourse TreeBank 2.0. In Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08), Marrakech, Morocco. European Language Resources Association (ELRA).

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. Preprint, arXiv:2305.18290.

Arndt Riester. 2019. Constructing QUD Trees, pages 164 – 193. Brill, Leiden, The Netherlands.

Craige Roberts. 2012. Information structure in discourse: Towards an integrated formal theory of pragmatics. Semantics and Pragmatics, 5(6):1–69.

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

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. Preprint, arXiv:2402.03300.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2024. Scaling llm test-time compute optimally can be more effective than scaling model parameters. Preprint, arXiv:2408.03314.

Weiwei Sun, Lingyong Yan, Xinyu Ma, Shuaiqiang Wang, Pengjie Ren, Zhumin Chen, Dawei Yin, and Zhaochun Ren. 2023. Is ChatGPT good at search?

investigating large language models as re-ranking agents. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pages 14918–14937, Singapore. Association for Computational Linguistics.

Ashima Suvarna, Xiao Liu, Tanmay Parekh, Kai-Wei Chang, and Nanyun Peng. 2024. QUDSELECT: Selective decoding for questions under discussion parsing. In Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 1288–1299, Miami, Florida, USA. Association for Computational Linguistics.

Nicolaus Tideman. 1995. The single transferable vote. Journal of Economic Perspectives, 9(1):27–38.

Jan Trienes, Sebastian Joseph, Jörg Schlötterer, Christin Seifert, Kyle Lo, Wei Xu, Byron Wallace, and Junyi Jessy Li. 2024. InfoLossQA: Characterizing and recovering information loss in text simplification. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 4263–4294, Bangkok, Thailand. Association for Computational Linguistics.

Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. 2024. Math-shepherd: Verify and reinforce LLMs step-by-step without human annotations. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 9426–9439, Bangkok, Thailand. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models. Preprint, arXiv:2201.11903.

Matthijs Westera, Laia Mayol, and Hannah Rohde. 2020. TED-Q: TED talks and the questions they evoke. In Proceedings of the Twelfth Language Resources and Evaluation Conference, pages 1118–1127, Marseille, France. European Language Resources Association.

Yangzhen Wu, Zhiqing Sun, Shanda Li, Sean Welleck, and Yiming Yang. 2025. Inference scaling laws: An empirical analysis of compute-optimal inference for problem-solving with language models. Preprint, arXiv:2408.00724.

Yating Wu, Ritika Mangla, Alexandros G. Dimakis, Greg Durrett, and Junyi Jessy Li. 2024. Which questions should i answer? salience prediction of inquisitive questions. Preprint, arXiv:2404.10917.

Yating Wu, Ritika Mangla, Greg Durrett, and Junyi Jessy Li. 2023a. QUDeval: The evaluation of questions under discussion discourse parsing. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pages 5344–5363, Singapore. Association for Computational Linguistics.

Yating Wu, William Sheffield, Kyle Mahowald, and Junyi Jessy Li. 2023b. Elaborative simplification as implicit questions under discussion. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pages 5525–5537, Singapore. Association for Computational Linguistics.

Tian Xie, Zitian Gao, Qingnan Ren, Haoming Luo, Yuqian Hong, Bryan Dai, Joey Zhou, Kai Qiu, Zhirong Wu, and Chong Luo. 2025. Logic-rl: Unleashing llm reasoning with rule-based reinforcement learning. Preprint, arXiv:2502.14768.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, and 16 others. 2025. Dapo: An open-source llm reinforcement learning system at scale. Preprint, arXiv:2503.14476.

Lunjun Zhang, Arian Hosseini, Hritik Bansal, Mehran Kazemi, Aviral Kumar, and Rishabh Agarwal. 2025. Generative verifiers: Reward modeling as next-token prediction. In The Thirteenth International Conference on Learning Representations.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023a. Judging LLM-as-a-judge with MT-bench and chatbot arena. In Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E Gonzalez, and Ion Stoica. 2023b. Judging llm-as-a-judge with mt-bench and chatbot arena. In Advances in Neural Information Processing Systems, volume 36, pages 46595–46623. Curran Associates, Inc.

Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2020. Fine-tuning language models from human preferences. Preprint, arXiv:1909.08593.

A Terminology and notation

Term	Definition
QUD instance (q)	a (typically implicit) question that links two parts in a piece of discourse. In practice, q is generated given the following inputs – that are drawn from the piece of discourse: (i) the context; (ii) an anchor, and (iii) an answer. To be useful in a QUD-based representation of the piece of discourse, the QUD q has to be carefully crafted to meet certain linguistic constraints with respect to each of these inputs.
Context (C)	this is the established ground (i.e. information that has already been encountered in the discourse up to, but not including, the Anchor anc).
Anchor (anc)	this is a sentence in the discourse that directly evokes (i.e. anchors) a given q.
Answer (ans)	this is a sentence further down in the discourse from anc, whose focus (i.e. its main part) answers q. ans must neither (i) be the same as anc, nor (ii) come before anc.
Focus	(also termed "at-issue content"); this is the main part of a sentence. i.e. it is distinct from the secondary/background information in a sentence.
Discourse-Old	this describes concepts (entities, events, or states) that have already been raised in C or anc.
Mediated	this describes concepts that have not been directly raised in C or anc, but are generally known or inferrable from one of the concepts that has already been raised so far (i.e. "discourse-old").
Answer Compatibility (*Comp.*)	assessment of whether ans appropriately answers q.
Givenness (*Givn.*)	assessment of whether q contains only information (about entities, events, or states) that has been raised up to the point of C and anc in the discourse, or could reasonably be inferred/come to the mind of a reader.
Anchor Relevance (*Relv.*)	assessment of whether q is likely to be raised by a person having read up to the point of C and anc. q should only contain information found in anc.

Table 7: Terms used in this paper, together with notations and definitions; based on guidelines descriptions in (Wu et al., 2023a)

B Data processing details

B.1 QUDEval

During the construction of QUDEval, for a given (C, ans) the anchor sentence anc in QUDEval may differ depending on the QUD construction approaches/models (namely, LLM-prompting with GPT4, ChatGPT and Alpaca 7B as well as the model from (Ko et al., 2023)) used to collect the q for it. As such, a portion of (C, anc, ans) in QUDEval only have output from a single approach/-model (i.e. we cannot compose a ranking task instance for it). For our study in Part I (Section 4), we are able to make use of the (C, anc, ans) that have q outputs (Q) from two or more approaches/models (we refer to one (C, anc, ans, Q) as a "rankable instance"). However, there are also cases in QUDEval where, for a given criterion, a rankable instance has QUDs that are all of the same score – we exclude such instances since the `RANK` and `RANK-PANEL` approaches invariably introduce a ranked order to the set of QUDs, and ranking with such cases would not be meaningful.

B.2 TEDQ

Since not all of the questions in TEDQ meet the criteria of a valid QUD, we have to identify the QUDs and extract them for use in our work. Specifically, we keep only questions in the TEDQ data release[17] that: (i) have been successfully marked with a question type; (ii) have been given the maximum answerability score of five by the annotators, and (iii) where the question ends with a "?". We also follow the setting in ***DCQA*** and keep only (C, anc, ans) tuples that fall within the first 20 sentences of the discourse. This gives us a set of (C, anc, ans) tuples we know that an answerable question can be formed and can serve as a QUD. Note that, unlike ***DCQA***'s sentence-level annotation, TEDQ was collected by showing annotators segments of the transcript comprised of two sentences or so (sentences longer than 150 words were split at punctuation points). Although annotators were made to select a short span of text 10 words or less that triggers a question, and we could in practice recover the sentence that encloses the annotated span (this would allow us to stay close to the DCQA and QUDEval view of QUDs as being triggered and answered by singular sentences), the nature of the type of discourse (transcript of monologues) is such that many sentences in TEDQ are very short (five words or less) and therefore have insufficient context to act as an anchor for a meaningful question. We therefore retain the original TEDQ segmentation strategy for the anchors anc. On the other hand, we narrow the answers ans to the sentence that en-

[17] `https://github.com/amore-upf/ted-q/blob/master/TED-Q_elicitation.csv`

closes the highlighted span so that the answer stays close to the DCQA and QUDEval view of ans for QUDs. This is important because a good QUD should only be answered by the focus of the ans, and having longer than necessary ans in generation and evaluation likely impacts their quality.

C Part I: Approaches for QUD automatic evaluation

▪ **(1) `RB-NLI`** is the set of rules-based and NLI measures used by (Suvarna et al., 2024) in the final step for their QUD construction model, which they use to select the QUD to return from the set of over-generations produced for different anc candidates to a given ans.[18] This approach involves the following for each of the criteria: **[1]** ▸ *Comp.*– using a natural language inference (NLI) model (`facebook/bart-large-mnli`), where they take the model's probability prediction that q is entailed by anc as the measure; **[2]** ▸ *Givn.*– using lexical matching, they take the overlap between the lemmas of the content words of q ($lemma_Q$) and the article up to and including anc ($lemma_{ctx}$) as the measure, and **[3]** ▸ *Relv.*– using lexical matching and similar to method for *Givn.*, they take the overlap of the lemmas of the content words of anc ($lemma_{anc}$) and that of the focus of q as the measure. For the latter, they assume that the focus of q is the maximum noun phrase within it. Note that the `RB-NLI` approach scores each QUD singly and our evaluation examines the ranking abilities of the approaches. Therefore, for a given ranking instance, we derive a ranking order based on the scores obtained for each q in it.

▪ **(2) `JDG`** LLM-as-judge is the prompting-based use of LLMs to obtain judgement about the quality of some text (Zheng et al., 2023b). (Wu et al., 2023a) found that using LLM-as-judge (which we refer to as `JDG`) for ref-free QUD evaluation gave close-to or higher macro-F1 score than the human upperbound for for *Comp.* & *Relv.*. We reuse the code released by (Wu et al., 2023a), but update from their use of GPT-4 to a more recent release of GPT (GPT4o), and we refer to this as `JDG` (GPT4o). Since this approach involves querying the LLM to give a score for each QUD in a zero-shot manner and since the remainder of the approaches we explore include few-shot prompting, we extended (Wu et al., 2023a)'s code for few-shot prompting too (which we refer to as $\texttt{JDG}_{fs}$). Similar to `RB-NLI`, here we derive a ranking order based on the scores obtained for each q in it.

▪ **(3) `RANK` & (4) `RANK-PANEL`** We first examine LLM-as-judge-ranker (which we refer to as `RANK`) for the task of ranking QUDs using a single LLM with Llama (8B), Qwen (7B) and GPT4o; we refer to these as `RANK` (Qwen 8B), `RANK` (Llama 7B) and `RANK` (GPT4o). We then go on to examine using a panel of LLM-as-judge-ranker comprising different compact LLMs of between 7B and 14B parameters (we use gemma-2-9b-it (9B), Llama-3.1-8B-Instruct (8B), phi-4 (14B) and Qwen2.5-7B-Instruct (7B); where we aggregated the preferences of the LLM-as-judge-rankers to return the ranking (which we refer to as `RANK-PANEL`). We follow (Han and Gardent, 2025) and use Single Transferable Vote (Tideman, 1995) to aggregate the preferences of the LLM-as-judge-rankers.

▪ **(5) `LLMQA (logp)`** Here, we prompt (see example of prompt in Appendix H) a decoder-only non-instruction fine-tuned LLM (Llama-3.2-3B) in a question answering (QA) task set-up. The model is given C, anc, q and all of the sentences in the article (as answer candidates ans_{cand}). We then obtain the log probabilities of each ans_{cand} as continuations of the input (i.e. the generated answer to q) and compute a score for *Comp.*. By using the sum of the log probabilities obtained for each ans_{cand}, we can induce a ranking for them. We use the normalised inverse rank position of the gold ans as the score for the QUD.[19]

C.1 Prompting strategies

In this section, we describe the prompting strategies relevant to approaches **(2)**, **(3)** and **(4)** above (see also Section 4). These are specifically in relation to `(2) JDG` (GPT4o), `(2)` $\texttt{JDG}_{fs}$ (GPT4o), `(3) RANK` (Llama 8B), `(3) RANK` (Qwen 7B), and `(4) RANK` (GPT4o) in Tables 10 & 11.

Few-shot prompting For the LLM-based approaches – except `JDG` (GPT4o) and `LLMQA (logp)`), we use three-shot prompting and explore two strategies: (i) vanilla prompting, and (ii) chain-of-thought (CoT)/**SCRS**

[18] We use their code, which can be found here: https://github.com/asuvarna31/qudselect/blob/main/selective_decoding/rule_based_approaches.py

[19] To give a worked example: assuming the eighth sentence of the 12-sentence article is the gold ans, and its log probability gives it a rank of three out of 12 answer candidates, the score is (12-3)/12 = 0.75.

(see Section 4). In (i), the exemplars only contain the task prompt, and the LLM's expected response is the rank order of the QUD candidates given to it. In (ii), the LLM's expected response is a reasoning chain covering each of the candidates before the rank order is given. The task prompt to the model includes a short description of the QUD framework, a set of useful definition of terms (e.g. what amounts to a context, what is considered "Mediated" etc), a scoring scheme for the criteria that is being assessed as well as a decomposition of the scoring & ranking assessment in the CoT (see example of few-shot prompt in Appendix F).

Obtaining few-shot exemplars To obtain the three exemplars for prompting, we started by searching the QUDEval dataset for (C, anc, ans) tuples that come with a set of four QUDs (Q) and where every q shares the same score for every one of the criteria (recall from above that these are not used in our evaluation data (Table 1)). For each $q \in Q$, we randomly assign a new score (with at most 1 repeat score in Q, since the scores are between 1 to 3 and there are 4 q). If the new score is different from the original score, one of the authors (native proficiency in English) modified the q to give q' so that it meets the new score according to the QUDEval guidelines (i.e. by either making it worse or improving it). For **SCRS**: At the same time, a rationale for why q' has the new score is written. By doing this for every $q \in Q$ for every criterion, we obtain the set of three exemplars we can use for few-shot CoT. The exemplars were used in this part for assessing the different evaluation approaches, and it was also used in Part II when using `RANK` (GPT4o) as the reward model and as evaluation measure.

C.2 Other metrics for assessing QUD evaluation approaches

Net Discounted Cumulative Gain (NDCG) is a measure used widely in information retrieval. In IR, where a query is paired with a set of documents that are each given a relevance weighting, NDCG measures how highly placed the relevant documents are within the prediction set returned. In our case, a (C, anc, ans) tuple is the query, and the set of QUDs is the document. We use the scores in QUDEval as the "relevance weighting" for computing the metric. We report NDCG@k (where k is the number of candidates in the ranking set).

Top-Rank Accuracy is a modified version of Exact Match. Whenever one (or more) of an approach's top-ranked QUD(s) falls in the set of top-ranked QUD(s) based on the human annotations, we count it as a match. In cases where an approach (e.g. `RB-NLI` and `JDG`) returns all candidates with the same score, we randomly pick one candidate as the winning one before computing the two measures. This avoids overcounting for such cases. The results for NDCG and Top-Rank Accuracy can be found in Tables 10 & 11.

D Implementation details

D.1 Knowledge distillation for reward model

D.1.1 Data

To obtain the data for knowledge distillation, we carried out a round of GRPO post-training of the Llama-3.2-3B-Instruct checkpoint.[20] We use LoRA (Hu et al., 2022) with a rank of 32 and an alpha ratio of 0.5 with dropout of 0.05, we set the sampling temperature to 1.0 to encourage exploration in the trajectory generations. The model after this round of post-training is discarded, but we retain the data generated through it for use in our knowledge distillation for the RM. The steps to obtain the data are as follows:

First, we run a "cold-start" stage with 2,000 steps through the ***DCQA*** training split (only the (C, anc, ans) from there) using rewards computed with the `RB-NLI` methods for the *Givn.* and *Relv.* criteria together with the `LLMQA (logp)` method for the *Comp.* criteria. We used the `LLMQA (logp)` method because we found that it gave ranking and scoring performance closer to human judgements (see Tables 10 & 11) compared to the original NLI method used by (Suvarna et al., 2024). Since the base Llama model has not been explicitly trained for generating QUDs, the purpose of this "cold-start" is to allow the LLM to learn to emit candidate QUDs (trajectories) that are of a more reasonable quality before we start collecting **SCRS** responses for them using `RANK` (GPT4o). This is inspired by the "cold-start" stage (albeit with SFT) that was taken in the post-training for the DeepSeek-R1 model (DeepSeek-AI et al., 2025).

Next, we continue to run the GRPO post-training for another 6,000 steps through the ***DCQA*** training split, but switch to using `RANK` (GPT4o) to compute the rewards. We use few-shot **SCRS** prompt-

[20] https://huggingface.co/meta-llama/Llama-3.2-3B-Instruct

ing here, and collect the rationales and scores returned by GPT4o. We call this data KD, which is comprised of 6,000 sets of Q, each of which is comprised of four candidate QUDs for a given (C, anc, ans) tuple. Each Q is accompanied by the set of **SCRS** output (i.e. scores and rationales) from RANK (GPT4o) for every one of the three criteria. The cold-start stage required two L40 GPUs (one for the LLM being post-trained, and another for the Llama-3.2-3B[21] model used for LLMQA (logp)). The second phase required a single L40 GPU (one for the LLM being post-trained) with asynchronous API calls to GPT4o to allow scoring for each of the criteria to be run simultaneously, and was completed in under 24 hours.

The set of reward functions we used in this data generation step – to obtain the Q – can be found in Table 8. We use a weight of 1.0 on the rewards from the QUD criteria reward functions and 0.5 otherwise.

Resp. format 1	0.5 if response strictly meets the format, 0.0 otherwise.
Resp. format 2	0.5 if response broadly meets the format, 0.0 otherwise.
Resp. format 3	0.125 for exactly 1 occurrence for each of these tags: <think>, </think>, <answer>, </answer>.
Resp. format 4	0.5 if the answer portion of the response does not contain any XML tags.
Thought length	0.5 if # words in think portion of response is between 250 and 350. 0.25 for 200-250 or 350-400, 0.125 for 150-250 or 400-450. 0 otherwise.
QUD length	0.5 if # characters in the QUD generated are between 7 and 15. 0.0 if less than 3 or more than 40. 0.125 otherwise.
Comp.	score from GPT4o for the criteria (i.e. 1,2 or 3) normalised to up to 1.0 (i.e. divided by 3).
Givn.	score from GPT4o for the criteria (i.e. 1,2 or 3) normalised to up to 1.0 (i.e. divided by 3).
Relv.	score from GPT4o for the criteria (i.e. 1,2 or 3) normalised to up to 1.0 (i.e. divided by 3).

Table 8: Reward functions and settings used for GRPO in our knowledge distillation data generation step. Additionally, for trajectories that are the empty string or does not end with a "?", we set all their rewards to zero.

D.1.2 Using supervised fine-tuning (SFT)

We use the data collected above (KD) together with sequence-level knowledge distillation (Kim and Rush, 2016) to obtain smaller more resource-efficient RMs (compared to using GPT4o). Specifically, we mix all the data points from KD for every critieria (i.e. 3 x 8,000 or a total of 18,000 data points) and filter out those with scores/rationales missing for the Q set.[22] About 600 data points were filtered out, leaving ∼17,400 data points (KD_{filter}). We use KD_{filter} and carry out supervised fine-tuning (SFT) on two LLMs: (i) Llama-3.2-3B-Instruct; and (ii) Qwen2.5-3B-Instruct.[23] We fine-tune for three epochs over the data, with batch size of 2, gradient accumulation step of 4 and a learning rate of 1e-4. We used LoRA with a rank of 128 here, and an alpha ratio of 0.5 with dropout of 0.05. Each of these SFT models took about 12 hours to train on a single H100 GPU. We found the Llama SFT RM to have large amounts of degenerate **SCRS** outputs (i.e. instances where we could not fully parse and obtain the rationale and score for every candidate) and slightly poorer macro F1 scores compared to the Qwen SFT RM (RM^{Qwen}_{SFT}). Thus we eventually used the Qwen 3B SFT model as the reward model for the next phase of GRPO post-training on QUD generation.

D.1.3 Using GRPO post-training

We also explored the use of GRPO post-training to obtain the RM. We did this by "cold-starting" from a checkpoint of RM^{Qwen}_{SFT} that had seen the first 4,000 data points of KD_{filter} and used reward functions that sought to have the model learn to generate scores as close to the ones in KD_{filter} (Table 8). We ran GRPO through another 12,000 data points of KD_{filter}. Training for the GRPO setup took substantially longer, and we did not reach the same number of steps over the KD data (i.e. slightly below one epoch instead of three epochs).

D.2 GRPO post-training for QUD generation

For the final GRPO post-trained QUD generation models, we use a similar approach to the one we used in Section D.1.1 for obtaining the knowledge distillation data. The main difference is that we use a refined set of reward functions that: (i) encourage more concise QUDs, and (ii) discourage the generation of certain terms specific to the prompt

[21] https://huggingface.co/meta-llama/Llama-3.2-3B

[22] Although we carried out the cold-start stage, and because of the exploratory nature of RL training, some of the q candidates were far from acceptable QUDs and we found that in spite of the few-shot prompting, GPT4o failed to return scores and rationales for some of these.

[23] meta-llama/Llama-3.2-3B-Instruct and Qwen/Qwen2.5-3B-Instruct on https://huggingface.co

instructions. The set of reward functions used in this stage can be found in Table 9. We use LoRA (Hu et al., 2022) with a rank of 32 and an alpha ratio of 0.5 with dropout of 0.05, we set the sampling temperature to 1.0 to encourage exploration in the trajectory generations. Here, for each set of trajectories sampled (i.e. at every sampling step of the training), we also iterated four times when optimising, which is a method to maximise learning efficiency per set of trajectories generated.

Resp. format 1	0.5 if response strictly meets the format, 0.0 otherwise.
Resp. format 2	0.5 if response broadly meets the format, 0.0 otherwise.
Resp. format 3	0.125 for exactly 1 occurrence for each of these tags: <think>, </think>, <answer>, </answer>.
Resp. format 4	0.5 if the answer portion of the response does not contain any XML tags.
Thought length	0.5 if # characters in think portion of response is between 250 and 350. 0.25 for 200-250 or 350-400, 0.125 for 150-250 or 400-450. 0 otherwise.
QUD length	0.75 if # words in the QUD generated are between 7 and 10. 0.5 if between 5 and 7 words or between 10 and 12 words. 0.0 otherwise
Excluded words	0.5 base, with -0.125 for every appearance of the following phrases: "the context", "the anchor", "the answer", "CTX", "ANS', "ANC", in the answer portion of the response.
Comp.	score from GPT4o for the criteria (i.e. 1,2 or 3) normalised to up to 1.0 (i.e. divided by 3).
Givn.	score from GPT4o for the criteria (i.e. 1,2 or 3) normalised to up to 1.0 (i.e. divided by 3).
Relv.	score from GPT4o for the criteria (i.e. 1,2 or 3) normalised to up to 1.0 (i.e. divided by 3).

Table 9: Reward functions and settings used for GRPO in our GRPO QUD generation training step. Additionally, for trajectories that are the empty string or do not end with a "?", we set all their rewards to zero.

We also included the use of refinements to the DeepSeek GRPO recipe that recent research have found helpful for improving the stability of training and eventual model performance. Specifically, we use the formulation proposed by (Liu et al., 2025a) for computing the trajectories' group advantage; by removing the original reward standard deviation denominator there, we avoid a potential bias that could lead to lower-reward trajectories having a longer length. Following (Yu et al., 2025), we also exclude truncated trajectories (i.e. degenerate outputs) from the advantage computation, which has been shown by to improve training stability; as well as separated the upper (0.28) and lower (0.20) bounds for the epsilon term used to maintain the updates to the model within a region close to the existing policy (similar to the "trust region" from the PPO objective).

Here, we used a single H100 GPU to host the LLM (Llama 3B or Qwen 3B) being post-trained, and three RTX4500 GPUs to host separate instances of RM^{Qwen}_{SFT} to obtain reward signal for the *Comp.*, *Givn.*and *Relv.* criteria. We use aysnchronous communications between the GPU instances for communicating the following between them: (i) the set of $\hat{q}$ from the trajectories sampled at every training step, which are sent to each RM; and (ii) the scores given by each criteria RM for the $\hat{q}$, which are returned to the master node hosting the LLM being post-trained.

For each of the Llama and Qwen models, we ran GRPO post-training with a "cold-start" of 1,000 (C, anc, ans) tuples from the ***DCQA*** training split using the `RB-NLI` & `LLMQA (logp)` methods to obtain reward signal. We followed this with training over another 4,000 (C, anc, ans) tuples from the ***DCQA*** training split using `RANK` (GPT4o). The whole post-training process took about 40 hours for each of the Llama and Qwen models.

D.3 `Classifier` for ref-free QUD evaluation

Note that the `Classifier` models are trained on a randomised split (by articles) of the QUDEval data. Since our test data – $\boldsymbol{QE}_{rank}$ – is drawn from across QUDEval, we could not use `Classifier` over our experiments in Part I (Section 4). Nonetheless, we can apply them in Part II (Section 5) – on the outputs of the baselines and $\mathbf{G_{rm}}$.

We follow the recipe described by (Suvarna et al., 2024) in their paper and use the training split of the QUDEval data they released under "oversample_qudeval".[24]. Their recipe calls for the use of Longformer-base (Beltagy et al., 2020) (for *Givn.*) and roBERTa-large (Liu et al., 2019) (for *Comp.* and *Relv.*) fine-tuned on text classification tasks, which is to predict the QUDEval scores given a QUD q and one of the following: C (*Givn.*), anc (*Relv.*), or ans (*Comp.*).

We find however, that there is a limitation in the original `Classifier` set-up for *Givn.* – (Suvarna et al., 2024) states that: *"For givenness, the input is the context (sentences before the anchor sentence in the article) and the question, and the answer is one of the three labels No-New., Ans-leak., and Hallu."* We note, however, that if the input does not include the answer sentence ans, it would be difficult for a model to make a reliable prediction on the "Ans-leak" and "Hallu" label. As such, we modified the recipe for the *Givn.* `Classifier` to add ans in the input. As the data released by (Suvarna et al., 2024) for *Givn.* does not contain the ans, and there are repeated q with different scores in the *Comp.* data they released (where q and ans are present), we could not confidently recover the ans for the data they released. For this reason, we took the following steps to draw the ans information using QUDEval and ***DCQA***. To stay as close to their recipe as possible, we: (i) filtered QUDEval for the subset that have the questions in (Suvarna et al., 2024)'s *Givn.* "train.csv", (ii) removed instances where the *Givn.* judgement was skipped, (iii) collected the C, anc, ans from ***DCQA*** for the remaining instances, and finally, (iv) randomly upsampled to the same number of instances as the *Givn.* "train.csv". We then fine-tune Longformer-base to predict the QUDEval score for these given (C, anc, ans).

[24] https://github.com/asuvarna31/qudselect/tree/main/automatic_evaluators/data/oversample_qudeval

E Distribution of RANK (GPT4o) scores – Part II

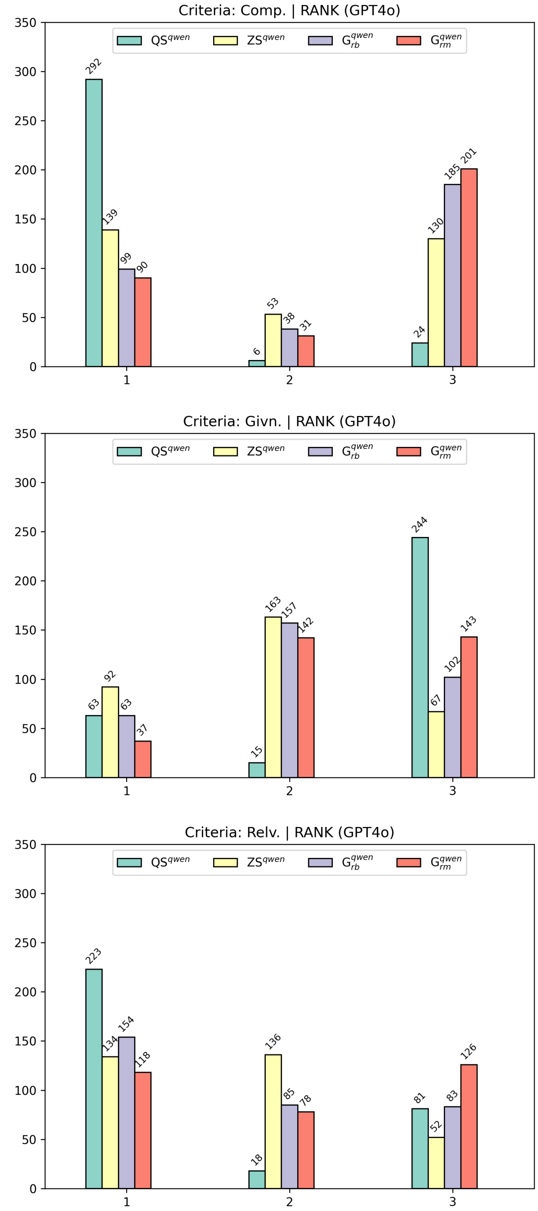

Figure 4: From top to bottom: $\boldsymbol{QE}_{rank}$, distribution of *Comp.*, *Givn.* and *Relv.* scores for **QS**, **ZS**, $\mathbf{G_{rb}}$ and $\mathbf{G_{rm}}$ (Qwen 3B).

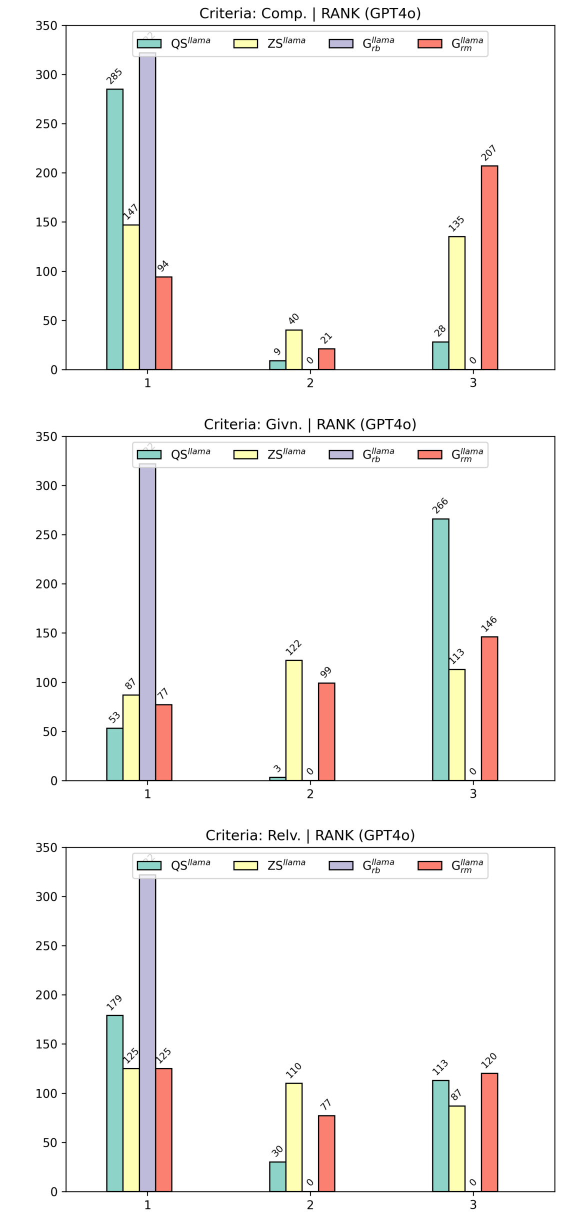

Figure 5: From top to bottom: QE_{rank}, distribution of Comp., Givn. and Relv. scores for **QS**, **ZS**, $\mathbf{G_{rb}}$ and $\mathbf{G_{rm}}$ (Llama 3B).

F Prompt example - SCRS

//

SCRS prompt for Answer Compatibility

<|begin_of_text|> <|start_header_id|> system <|end_header_id|>

Cutting Knowledge Date: December 2023
Today Date: 26 Jul 2024

You are an intelligent assistant that is as well-trained as a graduate linguistics student. You have the capabilities to understand a set of linguistic annotation instructions and can rank the quality of a set of Questions Under Discussion (QUD) instances for some specified Criteria and Scoring Scheme.<|eot_id|> <|start_header_id|> user <|end_header_id|>

I will provide you with 4 attempts that were made at generating a Questions Under Discussion (QUD) instance. Each attempt is indicated by a numerical identifier [] in front of it. Your task is to rank the attempts based on their quality with respect to the Criteria (using the Scoring Scheme) specified below.

###########
First of all, here are some important terminology and definitions to help you to understand the task:

Terminology and definitions:
- "Questions Under Discussion": this is a linguistic framework for representing discourse structure – a piece of discourse (for e.g. a news article) can be seen as a sequence of (implicit) questions that are successively posed and answered as the discourse progresses.
- "QUD": a QUD is an (implicit) question that links two parts in a piece of discourse. In practice, a QUD is generated given the following inputs that are drawn from the piece of discourse: (i) the context; (ii) an anchor sentence, and (iii) an answer sentence. To be useful in a QUD-based representation of the piece of discourse, the QUD has to be carefully crafted to meet certain linguistic constraints with respect to the context, anchor sentence and/or answer sentence (depending on which Criteria is being assessed).
- "context (CTX)": this is the established ground (i.e. information that has already been encountered in the discourse up to, but not including, the anchor sentence). The CTX can be empty (marked by a "[EMPTY]" symbol) if the anchor sentence is the first sentence in the discourse. - "anchor sentence (ANC)": this is a sentence in the discourse that directly evokes (i.e. anchors) the QUD. The QUD should be one that a person reading up to the point of the ANC in the discourse is likely to have come to mind (i.e. the next thing the person would be curious to read is the answer to the QUD).
- "answer sentence (ANS)": this is a sentence further down in the discourse from the CTX and ANC, and whose Focus (i.e. its main part) answers the QUD. The ANS (i) is a different sentence from that of the ANC, and (ii) never comes before the ANC.
- "Focus": this is the main part of a sentence (in our task here, either the ANC or the ANS). i.e. it is the most important new information that the speaker/writer wants to communicate to their audience. It is distinct from other background information in the sentence sentence. It is also termed as "at-issue" content.
- "Discourse-Old": this describes concepts (entities, events, or states) that have already been mentioned in the CTX or ANC.
- "Mediated": this describes concepts that have not been directly mentioned in the CTX or ANC, but are generally known or can be inferrable from one of the concepts that has already been mentioned (i.e. "Discourse-Old").

###########
Here is the criteria that you should judge the generated QUDs for:

Criteria: Answer Compatibility
Definition of Criteria: this assesses whether the QUD is appropriately answered by the answer sentence (ANS). The reason this Criteria matters is because, for a QUD to be useful, it has to be fully resolvable by the most important information of the ANS, which makes it a good link between the ANC and ANS.
Scoring Scheme:
- "direct and explicit": the Focus (and only the Focus) of the ANS answers the QUD. (3 points)
- "unfocused": the ANS contains the answer to the QUD; however, the answer is not the ANS's Focus. (2 points)
- "not answered": the ANS does not answer the QUD. (1 point)

Context: {... ... }
Anchor Sentence: {... ... }
Answer Sentence: {... ... }

QUD Instance Attempts:
[01] What is the international cricket council doing about it?
[02] Who are the parties alleged to have offered the bribes?
[03] When were the bribe attempts allegedly made?
[04] Which country's players were offered bribes to throw matches?

Rank the 4 attempts at QUD generation above. Rank them based on their quality with respect to the Criteria and the Scoring Scheme specified. All the QUD attempts should be included and listed using identifiers and in descending order of quality. If there are several attempts that are equally good amongst themselves, order their ranking by their index number (smallest first). The output format should be [] > [], e.g., [02] > [01]. It is very important that you give each score and rationale in a string that can be parsed into JSON. It is very important that you return the ranking results surrounded with '[START]' and '[STOP]'. Once you have responded with the ranking results, stop immediately; do not say anything else.<|eot_id|> <|start_header_id|> assistant <|end_header_id|>

First of all, I know that the higher the score the better, and the scores range from 1.0 to 3.0. When examining each QUD Attempt, the first thing I need to do is establish whether the QUD is answered by the ANS. I must be very careful and thorough doing this; be very precise and check whether the ANS properly answers the QUD. There are two possibilities at this point – either yes or no. If it is not, then I can immediately conclude that the QUD is a case of "not answered". If it is, then I need to go on to another step to establish if the part of the ANS that answers the QUD is the Focus of the ANS. There are again two possibilities at this point – either yes or no. If it is not (i.e. the answer to the QUD is not the Focus of the ANS), then it is a case of "unfocused"; if it is, then it is a case of "direct and explicit".Let's think step-by-step and assess each candidate first before ranking them. [S_COT] I think the candidates should get these scores based on the following assessments and reasoning:
[01] {"candidate": "[01]", "rationale": "here "it" in the QUD refers to the allegations of attempted bribery contained in the information that the Australian Cricket Board passed to the international cricket council (ICC). Since the ANS talks about how the ICC has launched an investigation into the allegations, therefore it does answer the QUD. To decide if the QUD is a case of "direct and explicit" or "unfocused", I will need to go on to assess whether this is the Focus of the ANS. I see that the ANC already mentions the allegations and that it involved leading players, so the new information being raised in the ANS are: (i) that the ICC launched an investigation, and (ii) the allegations were of bribery attempts to throw matches in the players' Nov/Dec tour in Pakistan. The latter merely serves to provide some date and location details about the alleged bribery, whereas the launch of the investigation is a more substantial piece of new information, thereby making it the Focus of the ANS. Based on this, I conclude that this QUD is a case of "direct and explicit".", "score": 3} .
[02] {"candidate": "[02]", "rationale": "i can see that the ANS does not have any information to resolve this QUD – it mentions that the bribes were allegedly made to leading Australian players, but ANS does not mention who was alleged to have offered the bribes. Therefore, this QUD is a case of "not answered".", "score": 1} .
[03] {"candidate": "[03]", "rationale": "the ANS mentions that the bribes were allegedly offered during the Australian players' tour in Pakistan in October-November, so it does answer the QUD. To decide if the QUD is a case of "direct and explicit" or "unfocused", I will need to assess if this information ("during their October-November tour of Pakistan") is the focus of the ANS. However, this is merely a minor detail about the alleged bribery; compared to the similarly new and more subtantial information in the ANS about the ICC launching an investigation, this is not the Focus of the ANS. Therefore, this QUD is a case of "unfocused". ", "score": 2} .
[04] {"candidate": "[04]", "rationale": "the ANS mentions that "Australian players" were allegedly bribed, therefore it does answer the QUD. To decide if the QUD is a case of "direct and explicit" or "unfocused", I will need to go on to assess whether this information ("Australian players") is the Focus of the ANS. While it is indeed new information that is first raised in the ANS, it is merely a minor detail about the alleged bribery; compared to the similarly new but more subtantial information in the ANS about the ICC launching an investigation, "Australian players" is not the Focus of the ANS. Therefore, this QUD is a case of "unfocused".", "score": 2} .
[E_COT]
Based on the above, I think the ranking should be as follows: [START] [1] > [4] > [3] > [2]
[STOP] <|eot_id|> <|start_header_id|> user <|end_header_id|>

{Exemplar # 2 ... ... }

{Exemplar # 3 ... ... }

Context: {... ... }
Anchor Sentence: {... ... }
Answer Sentence: {... ... }

QUD Instance Attempts:
[01] How likely has it been for species put on the endangered list to be subsequently saved from extinction?
[02] What does the successful increase in the number of Sierra Nevada bighorn sheep signify for endangered species?
[03] What other species have been able to avoid extinction?
[04] What was the cause of the bighorn sheep going to the brink of extinction?

Rank the 4 attempts at QUD generation above. Rank them based on their quality with respect to the Criteria and the Scoring Scheme specified. All the QUD attempts should be included and listed using identifiers and in descending order of quality. If there are several attempts that are equally good amongst themselves, order their ranking by their index number (smallest first). The output format should be [] > [], e.g., [02] > [01]. It is very important that you give each score and rationale in a string that can be parsed into JSON. It is very important that you return the ranking results surrounded with '[START]' and '[STOP]'. Once you have responded with the ranking results, stop immediately; do not say anything else.<|eot_id|> <|start_header_id|> assistant <|end_header_id|>

G Prompt example - GRPO post-training for QUD generation

QUD generation prompt

<|im_start|>system
A conversation between User and Assistant. The User describes a task and the Assistant solves it. The Assistant first thinks about the reasoning process in the mind and then provides the User with the answer. The reasoning process is enclosed within <think> </think> tags and the answer is enclosed within <answer> </answer> tags, (i.e. <think> reasoning process here </think> and <answer> answer here </answer>). The Assistant is as well-trained as a graduate linguistics student and has the capabilities to understand a set of linguistic annotation instructions, therefore the Assistant can write a question that meets the requirements of the Questions Under Discussion (QUD) framework. The QUD framework specifies a certain set of Criteria that the question must meet.

###########
Firstly, here are the set of Criteria you must follow to write a good QUD. It is very important that the QUD you write must score the maximum points on each of the Criteria.

Criteria: Answer Compatibility
Definition of Criteria: this assesses whether the QUD is appropriately answered by the answer sentence (ANS). The reason this Criteria matters is because, for a QUD to be useful, it has to be fully resolvable by the most important information of the ANS, which makes it a good link between the ANC and ANS.
Scoring Scheme:
- "direct and explicit": the Focus (and only the Focus) of the ANS answers the QUD. (3 points)
- "unfocused": the ANS contains the answer to the QUD; however, the answer is not the ANS's Focus. (2 points)
- "not answered": the ANS does not answer the QUD. (1 point)

Criteria: Givenness
Definition of Criteria: this assesses whether the QUD only contains concepts that (i) are in the context (CTX) or the anchor sentence (ANC) (i.e. "Discourse-Old"), or else (ii) the concepts are generally known, or can be easily inferred from concepts in the CTX and/or the ANC (i.e. "Mediated"). This Criteria matters because, for a QUD to be useful, it should not leak all or part of the answer, nor should it hallucinate information.
Scoring Scheme:
- "no new concepts": all of the concept(s) mentioned in the QUD is/are either "Discourse-Old" or "Mediated". (3 points)
- "answer-leakage": the QUD contains concepts that are neither "Discourse-Old" nor "Mediated". However, these new concepts can be found in the ANS, which is what distinguishes it from a case of "hallucination". (2 points)
- "hallucination": the QUD contains new concepts that are neither "Discourse-Old" nor "Mediated". Furthermore, these new concepts also cannot be found in the ANS, which is what distinguishes it from a case of "answer-leakage". (1 point)

Criteria: Anchor Relevance
Definition of Criteria: this assesses whether the QUD only/mostly contains information found in the ANC. The general idea of this Criteria is to establish whether a person could raise the QUD having just read the ANC (in other words, it is the most salient information up to this point of the discourse – i.e. from the ANC – that should be the trigger of a person's curiosity for the additional information to be sought, and the QUD should be formed to reflect this).
Scoring Scheme:
- "fully grounded": all/most of the content in the QUD follows from the ANC. (3 points)
- "partially grounded": only some content in the QUD is grounded in the ANC. (2 points)
- "not grounded": most/all of the content in the QUD is not from the

ANC. (1 point)

<|im_end|>
<|im_start|>user
Context: {... ...}
Anchor Sentence: {... ...}
Answer Sentence: {... ...}

Write a QUD given the CTX, ANC and ANS above. Make sure that the QUD you write scores maximum points on all of the Criteria specified. Start by thinking (it is very important to use the <think> </think> tags), then give the QUD in your answer (it is also very important to use the <answer> </answer> tags). Do not say anything else after your answer.
<|im_end|>
<|im_start|>assistant

H Prompt example: LLMQA (logp)

//

Prompt used in LLMQA (logp)

<|begin_of_text|> <|start_header_id|> system <|end_header_id|>

Cutting Knowledge Date: December 2023
Today Date: 26 Jul 2024

You are an intelligent assistant that can carefully analyse an article and a question that is related to it in order to answer the question.

I will give you a part of an article, followed by a question. I will then provide you with a number of potential candidates that will answer the question. Your task is to select the most appropriate answer candidate based on the contents of the article and the question. It is very important that you reply by repeating exactly (i.e. word-for-word) the candidate you select as the answer. Start your reply by giving the answer immediately, do not say anything else.

Article:
{context}

Question:
{qud}

Answer Candidates:
[01] {... ...}
[02] {... ...}
[02] {... ...}
.
.
.
[nn] {... ...}
Answer:
<|eot_id|> <|start_header_id|> assistant <|end_header_id|>

I Detailed results for Part I

Approach / Criteria	S	2-cands			3-cands			4-cands		
		Comp.	*Givn.*	*Relv.*	*Comp.*	*Givn.*	*Relv.*	*Comp.*	*Givn.*	*Relv.*
(1) `RB-NLI`	-	0.9442	0.9166	0.9484	0.9338	0.9059	0.9362	0.9417	0.9419	0.9577
	-	-	-	-	-	-	-	-	-	-
(2a) `JDG` (GPT4o) *	-	0.8974	0.9457	0.9506	0.8628	0.9091	0.9380	0.9057	0.9366	0.9549
	-	-	-	-	-	-	-	-	-	-
(2b) `JDG`$_{fs}$ (GPT4o)	-	0.9713	0.9355	0.9461	0.9770	0.9629	0.9483	0.9817	0.9467	0.9699
	Y	0.9713	0.9330	0.9529	0.9562	0.9079	0.9556	0.9779	0.9511	0.9643
(3) `RANK` (Llama 8B)	-	0.9351	0.9166	0.9271	0.9469	0.9266	0.9240	0.9521	0.9144	0.9173
	Y	0.9653	0.9229	0.9349	0.9612	0.9481	0.9361	0.9460	0.9159	0.9232
(3) `RANK` (Qwen 7B)	-	0.9563	0.9318	0.9293	0.9653	0.9115	0.9289	0.9692	0.9316	0.9188
	Y	0.9728	0.9242	0.9158	0.9752	0.9197	0.9196	0.9781	0.9305	0.9044
(3) `RANK` (GPT4o)	-	0.9804	0.9355	0.9529	0.9780	0.9263	0.9411	**0.9898**	0.9334	0.9540
	Y	0.9774	0.9532	0.9607	0.9827	0.9533	0.9593	0.9797	0.9414	0.9695
(3) `RANK` (Claude Opus 4)	-	0.9774	0.9671	**0.9787**	**0.9898**	**0.9674**	0.9684	0.9894	0.9719	0.9745
	Y	**0.9819**	**0.9684**	0.9663	0.9835	0.9619	**0.9723**	0.9891	**0.9818**	**0.9824**
(4) `RANK-PANEL`	-	0.9759	0.9229	0.9338	0.9681	0.9227	0.9198	0.9599	0.9184	0.9180
	Y	0.9789	0.9330	0.9293	0.9738	0.9427	0.9197	0.9617	0.9080	0.9137
(5) `LLMQA (logp)`	-	0.9683	-	-	0.9499	-	-	0.9514	-	-
	-	-	-	-	-	-	-	-	-	-

Table 10: **NDCG@k results**. For each approach, the first line is the base setting and the second line (with "Y" marked under the column S) is a setting with CoT/**SCRS** where applicable (i.e. this does not apply for settings such as RB-NLI or LLMQA (logp)). Underlined are the top-performing between no-CoT/**SCRS** and CoT/**SCRS** for each approach (i.e. this tells us whether using CoT/**SCRS** or not gives better performance). In bold are the top-performing overall. * denotes without the mapping function (Wu et al., 2023a) applied.

Approach / Criteria	S	2-cands			3-cands			4-cands		
		Comp.	*Givn.*	*Relv.*	*Comp.*	*Givn.*	*Relv.*	*Comp.*	*Givn.*	*Relv.*
(1) `RB-NLI`	-	0.6022	0.5225	0.5920	0.6286	0.4643	0.6667	0.7105	0.5455	0.6857
	-	-	-	-	-	-	-	-	-	-
(2a) `JDG` (GPT4o)*	-	0.7312	0.5135	0.5200	0.5857	0.6786	0.5873	0.8947	0.5152	0.6000
	-	-	-	-	-	-	-	-	-	-
(2b) `JDG`$_{fs}$ (GPT4o)	-	0.8495	0.4505	0.5760	0.9286	0.7679	0.7302	0.9211	0.6061	0.8571
	Y	**0.8817**	0.4955	0.6080	0.8857	0.7500	0.8095	0.9211	0.6970	0.8286
(3) `RANK` (Llama 8B)	-	0.5376	0.4054	0.4800	0.6429	0.5357	0.4762	0.7105	0.3636	0.3714
	Y	0.7527	0.4505	0.5360	0.7857	0.6607	0.5714	0.6316	0.4242	0.4857
(3) `RANK` (Qwen 7B)	-	0.6882	0.5135	0.4960	0.7714	0.4464	0.5238	0.8158	0.4848	0.4286
	Y	0.8065	0.4595	0.4000	0.8286	0.4464	0.4444	0.8684	0.4848	0.3429
(3) `RANK` (GPT4o)	-	0.8602	0.5405	0.6640	0.8714	0.5000	0.6349	**0.9474**	0.5152	0.7714
	Y	0.8387	0.6667	0.7200	0.9000	0.6607	0.7619	0.8158	0.6364	0.8571
(3) `RANK` (Claude Opus 4)	-	0.8387	0.7658	**0.8480**	0.8857	**0.7857**	**0.8571**	**0.9474**	**0.8485**	**0.9429**
	Y	0.8710	**0.7748**	0.7600	**0.9429**	**0.7857**	0.8095	0.9211	0.7879	0.8857
(4) `RANK-PANEL`	-	0.8280	0.4505	0.5280	0.7857	0.4821	0.4444	0.7632	0.3939	0.4286
	Y	0.8495	0.5225	0.4960	0.8571	0.6250	0.4444	0.7895	0.4242	0.4571
(5) `LLMQA (logp)`	-	0.7742	-	-	0.8286	-	-	0.7632	-	-
	-	-	-	-	-	-	-	-	-	-

Table 11: **Top-Rank Accuracy results.** * denotes without the mapping function (Wu et al., 2023a) applied.

J Comparison of vanilla LLM-as-judge-ranker and SCRS outputs

Criteria	Anchor Relevance
QUD candidates	[01] What else does Angelo like to do besides get the plastic loom? [02] What does Angelo like to do? [03] What are Angelo's interests or hobbies? [04] What does Angelo like to do for fun?

Strategy	Example response
vanilla LLM-as-judge-ranker	[01] > [02] > [04] > [03]
LLM-as-judge-ranker with **SCRS**	Let's think step-by-step and assess each candidate first before ranking them. I think the candidates should get these scores based on the following assessments and reasoning: {"candidate": "[01]", "rationale": "this QUD is asking for more activities Angelo does besides a specific mentioned activity he was interested in, but the idea of Angelo's general interests is mentioned in the ANC, which refers to him asking for a specific toy that interests him'a plastic loom that helps kids weave colorful rubber band bracelets.' This QUD references more aspects of what Angelo likes, so it is a case of'fully grounded.'", "score": 3} {"candidate": "[02]", "rationale": "this QUD is asking what Angelo likes to do (in general). Since the ANC discusses a toy that interests him and involves activities with rubber bands, it's related to what he might like to do. Therefore, this QUD is related and'fully grounded.'", "score": 3} {"candidate": "[03]", "rationale": "this QUD is asking for a list of Angelo's interests or hobbies. While this is similar to QUD [03], which mentions hobbies, the ANC mentions a specific toy (the loom), implying an'activity' rather than an'hobby', making the QUD somewhat less closely focused on the ANC's content. However, some information follows from the ANC. Therefore, this QUD is'partially grounded.'", "score": 2} {"candidate": "[04]", "rationale": "this QUD asks what Angelo likes to do for fun. The ANC mentions toys he is interested in but doesn't specifically mention fun activities or hobbies. The concept of what he likes overall follows from the mention of the specific toy. Therefore, while this QUD's information is found in the ANC, it covers a slightly broader scope, so it includes aspects that may not be directly mentioned but can be inferred to be implicitly stated. This is considered a case of'fully grounded.'", "score": 3} Based on the above, I think the ranking should be as follows: [01] > [02] > [04] > [03]

Table 12: Examples of LLM-as-judge-ranker outputs. Row below shows the structured JSON (in blue) rationale and score for each of the candidates being ranked. In red is the preferred order of the candidates.

K Part II: Scores for $\hat{q}$ on $\boldsymbol{TQ}_{ans}$

	Qwen			
Appr.	**QS**	**ZS**	$\mathbf{G_{rb}}$	$\mathbf{G_{rm}}$
Size	3B	3B	3B	3B
	RANK (GPT4o)			
Comp.	1.98	1.67	2.19	**2.26**
Givn.	**2.44**	1.80	2.19	2.30
Relv.	2.02	1.78	1.76	**2.15**
`Full`	6.44	5.24	6.13	**6.70**
	`Classifier`			
Comp.	2.17	2.56	**2.59**	2.54
Givn.	2.81	2.89	2.87	**2.91**
Relv.	**2.74**	2.43	2.59	2.70
`Full`	7.72	7.87	8.06	**8.15**
min.	5	3	7	7
mean.	8.0	24.7	12.4	14.2
max.	14	47	26	30
fail	9	0	0	0
same	16	0	0	0
	Llama			
Appr.	**QS**	**ZS**	$\mathbf{G_{rb}}$	$\mathbf{G_{rm}}$
Size	3B	3B	3B	3B
	RANK (GPT4o)			
Comp.	**2.56**	1.83	1.00	2.19
Givn.	**2.76**	1.76	1.00	2.24
Relv.	**2.52**	1.85	1.00	2.11
`Full`	**7.83**	5.44	3.00	6.54
	`Classifier`			
Comp.	2.56	**2.67**	1.07	2.44
Givn.	2.89	**2.98**	2.33	2.81
Relv.	**2.89**	2.65	1.41	2.44
`Full`	**8.33**	8.30	4.81	7.70
min.	5	6	8	5
mean.	8.7	28.0	10.7	17.0
max.	19	83	21	72
Fail	0	0	83	4
Same	15	0	0	4

Table 13: Scores for QUDs generated on (C, anc, ans) tuples in $\boldsymbol{TQ}_{ans}$. **Significance tests:** although the average global score (`Full`) of the $\mathbf{G_{rm}}$ outperforms **QS** for the Qwen 3B model, we only found statistical significance that $\mathbf{G_{rm}}$ is better than **QS** with `Classifier`.

L Part II: Scores for $\hat{q}$ on $\boldsymbol{QE}_{rank}^{human}$; subset with a human-written q collected

	Qwen							
Appr.	**QS**	**ZS**	$\mathbf{G_{rb}}$	$\mathbf{G_{rm}}$	**QS**	**ZS**	$\mathbf{G_{rb}}$	$\mathbf{G_{rm}}$
Size	3B	3B	3B	3B	7B	3B	3B	3B
	RANK (GPT4o)							
Comp.	1.22	1.97	**2.36**	2.34	1.27	1.93	**2.37**	2.33
Givn.	**2.52**	1.89	2.16	2.35	**2.44**	1.93	2.27	2.34
Relv.	1.49	1.76	1.78	**2.12**	1.67	1.77	1.82	**2.09**
`Full`	5.24	5.61	6.31	**6.80**	5.38	5.64	6.46	**6.76**
	`Classifier`							
Comp.	1.65	**2.80**	2.70	2.70	1.80	←	←	←
Givn.	**2.87**	2.66	2.72	2.73	**2.92**	←	←	←
Relv.	2.28	2.42	2.53	**2.63**	2.47	←	←	←
`Full`	6.86	7.84	7.89	**7.94**	7.20	←	←	←
min.	4	3	4	4	2	←	←	←
mean.	10.3	24.0	12.0	13.4	7.8	←	←	←
max.	40	55	25	38	15	←	←	←
Fail	4	0	0	0	1	←	←	←
Same	39	0	0	0	27	←	←	←
	Llama							
Appr.	**QS**	**ZS**	$\mathbf{G_{rb}}$	$\mathbf{G_{rm}}$	**QS**	**ZS**	$\mathbf{G_{rb}}$	$\mathbf{G_{rm}}$
Size	3B	3B	3B	3B	8B	3B	3B	3B
	RANK (GPT4o)							
Comp.	1.27	1.96	1.00	**2.26**	1.26	1.95	1.00	**2.29**
Givn.	**2.71**	2.10	1.00	2.20	**2.65**	2.01	1.00	2.18
Relv.	1.78	1.95	1.00	**2.00**	1.73	1.97	1.00	**2.05**
`Full`	5.76	6.01	3.00	**6.45**	5.64	5.93	3.00	**6.52**
	`Classifier`							
Comp.	1.83	**2.67**	1.20	2.61	1.89	←	←	←
Givn.	**2.92**	2.66	2.25	2.65	**2.93**	←	←	←
Relv.	2.44	2.47	1.51	**2.51**	2.47	←	←	←
`Full`	7.18	**7.83**	4.95	7.77	7.30	←	←	←
min.	5	2	1	5	4	←	←	←
mean.	9.0	23.7	10.3	17.1	9.5	←	←	←
max.	23	116	41	379	28	←	←	←
Fail	0	1	83	5	0	←	←	←
Same	35	0	0	0	51	←	←	←

Table 14: Scores for QUDs generated on subset of (C, anc, ans) tuples in $\boldsymbol{QE}_{rank}^{human}$ where a human-written q has been collected (i.e. in ***DCQA***). ← denotes that the same values as those in the corresponding set of columns on the left applies. **Significance tests:** underlined denote statistical significance (p-value < 0.05 for both one-tailed t-test with bootstrap resampling and Wilcoxon signed-test) that the approach is better than **QS**.

M Comparison of outputs QS vs G_{rm}

Context (CTX)	ST. PETERSBURG, Fla. - Andy Warhol was an early adopter of selfies, if a new exhibit showcasing his work is any indication. The iconic pop artist's work is showcased in a large new exhibit at The Dali Museum in St. Petersburg, Fla., and it includes paintings, drawings and yes, self-portraits - taken with the Instagram of the 1970s, Polaroid. There's Warhol posing with a skull, Warhol looking moody in black and white and Warhol wedged in between John Lennon and Yoko Ono (bonus points: celebrity selfie!). And despite the 1970s clothing and Studio 54-era glitter of the art and photographs, the exhibit feels fresh. Warhol's vision of pop culture and fame fits right in with America's 21st century love of the Kardashians and TMZ.com. The exhibit of more than 100 works and some films opens to the public Saturday and runs through April 27. The works are on loan from the Warhol Museum in Pittsburgh, where Warhol was born and raised before moving to New York City. It's a first for The Dali, which is an entire museum devoted to the surrealist master's work. Museum executive director Hank Hine said during a media tour Wednesday that the exhibit - called 'Warhol: Art. Mortality'. - is the first large-scale special exhibit for the museum since it moved into a stunning new building along St. Petersburg's waterfront in 2011. 'This is totally appropriate to the legacy of Dali,' said Hine. The exhibit showcases some of Warhol's famous silkscreens, including a red self-portrait taken a year before he died. There's also several panels of Jackie Kennedy Onassis and a few quirky versions of his Campbell's Soup cans. There are also TV monitors set up to show the artist's 'screen tests,' short films of a single person against a plain background. Salvador Dali and Warhol had much in common, Hine said, and the exhibit contains two photos of the men goofing around in a hotel room together. Dali was a generation older - born in 1904 - than Warhol, who was born in 1928, but both captivated the public with their attention-getting and sometimes controversial works of art. Warhol often visited Dali when Dali stayed in a suite at the St. Regis Hotel in New York (one story goes that Dali tied Warhol to a spinning board and splattered paint over him).
Anchor sentence (ANC)	Warhol also made one of his famous 'screen tests' of Dali, and like the others filmed, Warhol told Dali not to move during the three-minute filming.
Answer sentence (ANS)	Dali ignored Warhol's instruction to sit still and left the frame.

Model	**QS**(Qwen)		G_{rm}(Qwen)	
QUD	When did Andy Warhol take selfies?		What happened during the filming of Dali's 'screen test'?	
TTC output	-		The QUD should score maximum points on all the Criteria: Answer Compatibility, Givenness, and Anchor Relevance. To achieve this, I need to formulate a question that is directly answered by the ANS, contains no new concepts not in the Context or Anchor Sentence, and is grounded in the Context and Anchor Sentence.	
	Score	Rationale	Score	Rationale
Comp.	1	the ANS does not provide any information about when Andy Warhol took selfies. It only discusses Dali's reaction during the "screen test". Therefore, this QUD is a case of 'not answered'	3	the ANS answers this QUD by describing what happened during the filming of Dali's "screen test", specifically that Dali ignored Warhol's instruction and left the frame. This is the main focus of the ANS, making it a case of 'direct and explicit'
Givn.	3	this QUD mentions the following concept: (i) Andy Warhol taking selfies. The concept corresponding to "when" in the QUD is what's being queried for, so we do not need to consider it. The concept of Warhol taking selfies is mentioned in the CTX, so it is Discourse-Old. Therefore, this QUD is a case of 'no new concepts'	3	this QUD mentions the following concepts: (i) the filming of Dali's "screen test". The concept corresponding to "what happened during" in the QUD is what's being queried for, so we do not need to consider it. The filming of Dali's "screen test" is mentioned in the ANC, so it is Discourse-Old. Therefore, this QUD is a case of 'no new concepts'
Relv.	1	this QUD is asking when Andy Warhol took selfies. The ANC does not mention anything about Warhol taking selfies, so none of the content of this QUD follows from the ANC. Therefore, this QUD is a case of 'not grounded'	3	this QUD is asking what happened during the filming of Dali's "screen test". The ANC mentions that Warhol made a "screen test" of Dali and instructed him not to move, which is directly related to the QUD. Therefore, this QUD is a case of 'fully grounded'.

Table 15: Comparison of QUDs generated by the Qwen 3B versions of (i) **QS** and (ii) G_{rm}, our GRPO post-trained model. The top part of the table gives context, anchor and answer sentences the QUD should be written for. The lower half shows the generated QUDs and the scores and rationales obtained from RANK (GPT4o).

Context (CTX)	JERUSALEM - Professor Mohammed Dajani expected criticism when he took Palestinian students to Poland last month to visit the site of the Auschwitz concentration camp. But he wasn't prepared for the uproar that followed.
Anchor sentence (ANC)	In online posts and comments, Palestinian critics denounced the visit as treason.
Answer sentence (ANS)	Acquaintances counseled the professor to keep a low profile, stay away from his university campus and consider taking a vacation abroad, he recalled.

Model	**QS**(Qwen)		$\mathbf{G_{rm}}$(Qwen)	
QUD	Was he prepared for the uproar that followed?		What did acquaintances advise Professor Mohammed Dajani?	
TTC output	-		The given Context, Anchor Sentence, and Answer Sentence depict a situation where Professor Mohammed Dajani took Palestinian students to Auschwitz, leading to criticism and some advice from acquaintances. The QUD should focus on the advice given by acquaintances and be grounded in the Context and Anchor Sentence.	
	Score	Rationale	Score	Rationale
Comp.	1	the ANS does not address whether Professor Dajani was prepared for the uproar that followed, so this QUD is a case of 'not answered'.	3	the ANS explicitly mentions what acquaintances advised Professor Dajani to do, which is to keep a low profile, stay away from his university campus, and consider taking a vacation abroad. This is the Focus of the ANS, so this QUD is a case of 'direct and explicit'.
Givn.	1	this QUD mentions the following concepts: (i) Professor Mohammed Dajani, and (ii) his preparedness for the uproar. The concept corresponding to "was he prepared" in the QUD is what's being queried for, so we do not need to consider it. The first concept (i) is mentioned in the CTX and ANC, but the second concept (ii) is not mentioned there and cannot be easily inferred from the concepts already mentioned there. In order to decide if this is a case of 'answer-leakage' or 'hallucination', I need to check if this concept (ii) is mentioned in the ANS. I can see that it is not mentioned in the ANS. Therefore, this QUD is a case of 'hallucination'.	2	this QUD mentions the following concepts: (i) acquaintances, and (ii) their advice to Professor Mohammed Dajani. The concept corresponding to "what did they advise" in the QUD is what's being queried for, so we do not need to consider it. The first concept (i) is mentioned in the ANC, but the second concept (ii) is not mentioned there and cannot be easily inferred from the concepts already mentioned there. In order to decide if this is a case of 'answer-leakage' or 'hallucination', I need to check if this concept (ii) is mentioned in the ANS. I can see that it is mentioned in the ANS (i.e. "Acquaintances counseled the professor to keep a low profile..."), therefore, this QUD is a case of 'answer-leakage'.
Relv.	1	this QUD is asking whether Professor Mohammed Dajani was prepared for the uproar that followed. The ANC mentions that Palestinian critics denounced the visit as treason, which is part of the uproar, but it does not mention, or make reference to, whether Professor Mohammed Dajani was prepared for it. Therefore, this QUD is a case of 'not grounded'.	1	this QUD is asking about what acquaintances advised Professor Mohammed Dajani. The ANC does not mention, or make reference to, any advice given by acquaintances. Therefore, this QUD is a case of 'not grounded'.

Table 16: Comparison of QUDs generated by the Qwen 3B versions of (i) **QS** (ii) $\mathbf{G_{rm}}$.

N Comparison of reasoning outputs from G_{rm}(Qwen) and G_{rm}(Llama) models

Qwen	**Llama**
The QUD should only contain concepts from the context and anchor sentence (which are "Discourse-Old"), and it should be fully grounded, directly and explicitly answered by the answer sentence. It should not contain any new information and should not be answer-leakage. It should only assess salient information from the context, which is about the virus in honeybees causing colony collapse.	To write a QUD that scores maximum points on all of the Criteria, I need to ensure that the QUD is directly and explicitly answered by the answer sentence, only contains concepts that are in the context or the anchor sentence, and is fully grounded in the anchor sentence.
q: What problem are scientists attributing the collapse of colonies to currently?	q: What is the relative role of the virus in the demise of honeybee colonies?
The QUD should mostly follow from the anchor sentence, and the anchor sentence mainly talks about a recent international report on climate change not having much lyrical language while being long in technical details. So, the QUD focuses on getting an idea about the content of this document.	To write a QUD that scores maximum points on all of the Criteria, I need to make sure that the QUD is direct and explicit, only contains concepts that are in the context or the anchor sentence, and is fully grounded in the anchor sentence.
q: What does the document from the United Nations Intergovernmental Panel on Climate Change (IPCC) contain?	q: What is the main characteristic of the recent international report on climate change?
The provided answer sentence gives general information about the actions taken by federal police. The pertinent factual information to meet the criteria related to Answer Compatibility, Givenness, and Anchor Relevance is the first clause of the answer sentence, focusing on the federal police taking action.	To write a QUD that scores maximum points on all of the Criteria, I need to make sure that the QUD is direct and explicit, only contains concepts that are in the context or the anchor sentence, and is fully grounded in the anchor sentence.
q: How did federal police react to the state truce and rebel presence?	q: What is the president's next move after ending the truce with the leftist Indian rebels?
A thorough analysis of the given context (CTX), anchor sentence (ANC), and answer sentence (ANS) points to the following breakdown of criteria. The QUD needs to be anchored in the given context, have an appropriate Focus identified only in the ANS, and not introduce any answers, but must also be fully grounded in the ANC.	To write a QUD that scores maximum points on all of the Criteria, I need to make sure that the QUD is directly and explicitly answered by the answer sentence, only contains concepts that are in the context or the anchor sentence, and is fully grounded in the anchor sentence.
q: What has happened in the village?	q: What is the current number of refugees in the area?
The QUD should be fully grounded, answer-compatible, and given in "Discourse-Old" or "Mediated" concepts. The anchor sentence mentions a certain number of people who supported rebels, food left behind, and lights on.	To write a QUD that scores maximum points on all of the Criteria, I need to make sure that the QUD is directly and explicitly answered by the answer sentence, only contains concepts that are in the context or the anchor sentence, and is fully grounded in the anchor sentence.
q: How many people did Abraham Castaneda of the National Nutrition Institute mention supported Indian rebels?	q: What did Abraham Castaneda of the National Nutrition Institute say about the people who supported Indian rebels?

Table 17: Random samples of reasoning outputs and QUDs from G_{rm}(Qwen) and G_{rm}(Llama). G_{rm}(Llama) produces generic reasoning tokens which is simply a summary of the prompt instructions and are relatively similar across instances, whereas G_{rm}(Qwen) produces varied reasoning traces, which also summarises the prompt instructions but some of which refer to parts of the prompts including elements of the (C, anc, ans).