Learning to Explore and Select for Coverage-Conditioned Retrieval-Augmented Generation

Takyoung Kim[1,†] **Kyungjae Lee**[2] **Young Rok Jang**[2]
Ji Yong Cho[2,3] **Gangwoo Kim**[4,†] **Minseok Cho**[2] **Moontae Lee**[2,5]

[1]University of Illinois Urbana-Champaign [2]LG AI Research
[3]Cornell University [4]Korea University [5]University of Illinois Chicago
tk30@illinois.edu moontae.lee@lgresearch.ai

Abstract

Interactions with large language models (LLMs) often yield long and detailed responses, leveraging both parametric knowledge and retrieval-augmented generation (RAG). While these responses can provide rich insights, they often include redundant or less engaging content not aligned with user interests. This issue becomes apparent when users specify particular subtopics to include or exclude – termed **coverage-conditioned ($\mathbf{C^2}$) queries** – as LLMs often struggle to provide tailored responses. To address this challenge, we investigate the role of *query outlines*, sequences of subqueries designed to guide LLMs in generating responses that meet specific user requirements. To systematically create and evaluate these outlines, we introduce **QTREE**, a dataset of 10K hierarchical sets of information-seeking subqueries that define structured boundaries for outline creation and evaluation in C^2 scenarios[1]. Additionally, we develop **QPLANNER**, a 7B language model trained to generate customized outlines within boundaries of QTREE. We evaluate the effectiveness of the generated outlines through automatic and human judgements, focusing on their impact within retrieval-augmented generation (RAG) systems. Experimental results demonstrate that QPLANNER, especially when trained with alignment techniques like DPO, generates higher-quality outlines that better fulfill diverse user needs.

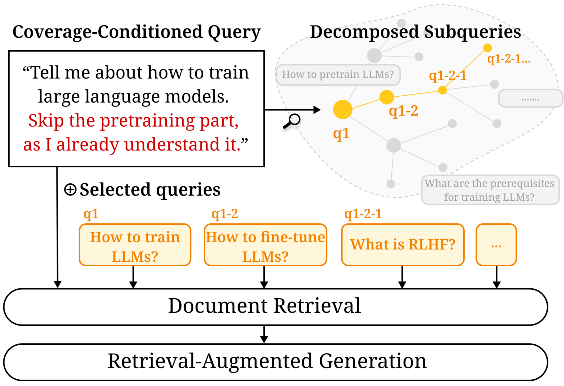

Coverage-Conditioned Query
"Tell me about how to train large language models. Skip the pretraining part, as I already understand it."
Decomposed Subqueries
How to pretrain LLMs?
q1-2-1...
q1-2-1
q1-2
q1
....
What are the prerequisites for training LLMs?
⊕ Selected queries
q1
How to train LLMs?
q1-2
How to fine-tune LLMs?
q1-2-1
What is RLHF?
...
Document Retrieval
Retrieval-Augmented Generation

Figure 1: QTREE constrains the range of available outlines for the user's C^2 query, and tailored outlines satisfying the requirement of C^2 query are selected for RAG downstream tasks.

1 Introduction

Recent advancements of large language models (LLMs) have enabled them to provide long and detailed responses by leveraging their parametric knowledge. As these models improve, human-machine interaction interfaces (*e.g.*, chat and acoustic interfaces) – which have been studied for a long time (Levin et al., 2000) – have become more sophisticated, allowing users to request highly specific and personalized information. Proprietary chat services such as ChatGPT (OpenAI, 2023), Gemini (Google, 2024), and BingChat have further accelerated the exploration of personalized information. Additionally, retrieval-augmented generation (RAG) methods are being adopted to enhance the relevance and timeliness of LLM responses by integrating external knowledge.

Despite these advancements, LLMs often struggle with delivering tailored responses when faced with complex user queries. For instance, a user might request LLMs to provide information on *Generative AI*, focusing specifically on its historical context while *excluding* recent trends. Crafting such meticulously composed responses is difficult for LLMs for two reasons: (1) LLMs' long-form outputs can contain innumerable combinations of relevant topics, and (2) there is no established gold standard for long-form text generation (Krishna et al., 2021; Xu et al., 2022, 2023). Recognizing this, we first define queries that constrain the information coverage of certain topics as **coverage-**

[†] Work done as a research intern at LG AI Research.

[1] Our resources are available at https://github.com/youngerous/qtree.

Findings of the Association for Computational Linguistics: NAACL 2025, pages 1460–1480
April 29 - May 4, 2025 ©2025 Association for Computational Linguistics

conditioned (C^2) queries, where "coverage" refers to the user's intent to instruct LLMs to include or exclude specific subtopics within their responses. These C^2 queries especially pose challenges in constructing long-form RAG responses as they require selective document retrieval as well.

To improve LLM responses for users' complex queries, there have been works on *query outlining*, creating sequences of intermediate subtopics to guide long-form responses. Query outlining has been effective in areas like long story generation (Fan et al., 2018; Sun et al., 2022; Yang et al., 2022, 2023; Wang et al., 2024; Shao et al., 2024). However, generating high-quality outlines that address complex queries like C^2 queries remains challenging, as there is no systematic approach for creating and evaluating such outlines.

With the concepts of C^2 query and query outlining in place, we pose two key research questions:

RQ1. How can we create and evaluate better outlines for C^2 *queries?*

RQ2. Can these outlines improve RAG systems by serving as search queries and content drafts?

To address *RQ1*, we present **QTREE**, a dataset comprising 10K hierarchical sets of information-seeking subqueries (with 39 subqueries in each set) that interpret user queries with diverse perspectives, facilitating the exploration and selection of appropriate outlines for C^2 queries. The hierarchies in QTREE are organized according to the abstraction level of the main topic, defining tangible boundaries of available outlines. For example, as illustrated in Figure 1, hierarchical subtopics related to processes after pretraining (*i.e.*, *Fine-tuning* and *RLHF*) are selected as proper outlines for RAG response among various viewpoints on the topic of *Training LLMs*, following the requirements of the C^2 query. In contrast, less relevant subtopics in QTREE (*e.g.*, *Pretraining LLMs*) will not be a desirable outline for the C^2 query. By leveraging QTREE, we can systematically create and judge outlines for long-form responses, ensuring that they align with the user's coverage constraints.

Regarding *RQ2*, we introduce **QPLANNER**, an autoregressive 7B language model designed to generate tailored outlines within QTREE's hierarchical boundaries. We hypothesize that high-quality outlines aligned with C^2 queries can improve both document retrieval and response generation in RAG systems. We also evaluate QPLANNER 's performance through both automatic metrics and human judgments, assessing the quality of the generated outlines and their impact on downstream tasks. Experimental results on C^2 queries from diverse domains (*i.e.*, Wikipedia and expert domains) demonstrate that training QPLANNER with preference alignment further improves both outline quality and overall RAG performance.

Our contributions are summarized as follows:

1. We present QTREE, a novel dataset of 10K hierarchical subquery sets that define boundaries for available outlines, facilitating the creation and evaluation of better outlines for coverage-conditioned (C^2) queries (addressing *RQ1*).

2. We introduce QPLANNER, an autoregressive language model designed to generate customized outlines that improve document retrieval and content generation in RAG systems (addressing *RQ2*).

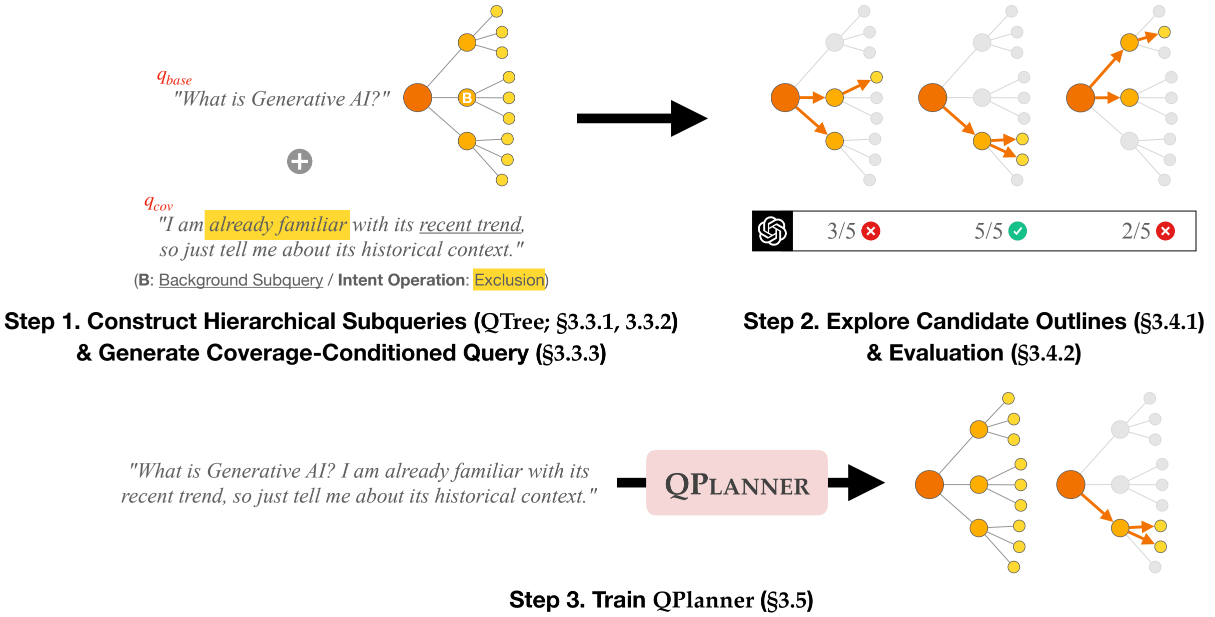
3. We conduct comprehensive evaluations, including automatic metrics and human judgments, to validate the effectiveness of our approach in enhancing outline quality and RAG performance.

2 Related Work

2.1 Query Modification with LLMs

Integrating retrieval systems with LLMs has become crucial, with query modification playing a pivotal role in improving information retrieval outcomes. Recent advancements focus on prompting LLMs to provide detailed information, such as expected documents or pseudo-answers, for query expansion (Wang et al., 2023; Jagerman et al., 2023). Furthermore, reward signals are being used to fine-tune query modification models, optimizing search results based on the ranking of retrieved documents (Ma et al., 2023; Yoon et al., 2024). Additionally, complex questions are being decomposed into simpler subqueries to enhance retrieval accuracy and response generation (Khot et al., 2023).

Among various query modification strategies, query outlining stands out as an effective method for generating long responses. Outlining techniques have been primarily studied in tasks such as story generation (Fan et al., 2018). Yang et al. (2022, 2023) have also emphasized the importance of outline for narrative generation, while Shao et al.


q_{base}
"What is Generative AI?"
B
q_{cov}
"I am already familiar with its recent trend, so just tell me about its historical context."
(B: Background Subquery / Intent Operation: Exclusion)
3/5
5/5
2/5
"What is Generative AI? I am already familiar with its recent trend, so just tell me about its historical context."
QPLANNER

Step 1. Construct Hierarchical Subqueries (QTree; §3.3.1, 3.3.2) & Generate Coverage-Conditioned Query (§3.3.3)

Step 2. Explore Candidate Outlines (§3.4.1) & Evaluation (§3.4.2)

Step 3. Train QPlanner (§3.5)

Figure 2: The overview of our framework. `[Step 1]` Base query (q_{base}) is decomposed into subqueries with diverse viewpoints (QTREE), preceded by generating coverage query (q_{cov}). `[Step 2]` After C^2 candidate outlines are extracted, a judge LLM evaluates each outline and selects the best-scored one. `[Step 3]` Utilizing this dataset, QPLANNER is trained to sequentially generate its own QTREE and preferred outline by taking the C^2 query as an input.

(2024) explored the outline as tools for presenting diverse perspectives through iterative conversational processes. More recently, Lee et al. (2024) improved free-form writing with outline augmentation. However, despite these advances, it has seen less attention in retrieval-augmented contexts. In addition, none of these studies systematically evaluate the generated outlines in complex scenarios (*e.g.*, C^2 scenarios). Our work aims to address this gap by proposing a controlled evaluation testbed for outlines and their impact on long-form responses.

2.2 Evaluation of Long-form Responses

Evaluating long-form responses from LLMs presents a significant challenge due to the subjective and multifaceted nature of the task. Previous studies (Krishna et al., 2021; Xu et al., 2022, 2023) highlight the limitations of automated metrics in accurately assessing long-form texts, underscoring the need for more nuanced evaluation methods. Several approaches have emerged to tackle this issue by incorporating multi-metric evaluation frameworks (Liang et al., 2023; Gehrmann et al., 2023; Shevlane et al., 2023; Ye et al., 2024), as well as task-specific metrics for fact verification and summarization (Min et al., 2023; Krishna et al., 2023). Recent research has also investigated model-based evaluations where learned models are used to generate automated scores (Yuan et al., 2021; Liu et al., 2023; Kim et al., 2024a).

While most of these studies focus solely on the evaluation of long-form responses, we extend this by evaluating both the outlines and responses they guide. Drawing from insights in cognitive psychology (Kellogg, 1988), we argue that outlines alleviate the cognitive overload for readers, functioning as effective content drafts and providing the core structure for long-form writing.

3 Framework

3.1 Background

We refer to QTREE as a tree-shaped hierarchical set of subqueries (defining "subquery" as each node in QTREE) derived from a single user query. We set both the depth and the width of QTREE at three levels (*i.e.*, 3+9+27=39 subqueries in each QTREE). Additionally, as illustrated in Figure 1, we define C^2 query as the concatenation of the user's original query (*base query*; q_{base}) and additional coverage-constraining query (*coverage query*; q_{cov}), represented as $C^2 = [q_{base}; q_{cov}]$.

3.2 Overview

Figure 2 illustrates the procedural framework, including the construction of QTREE and QPLANNER. Followed by collecting q_{base} (Section 3.3.1), we construct QTREE (Section 3.3.2) and generate

Dataset	Source	Train	Test
ASQA (Stelmakh et al., 2022)	Wikipedia	4,353	100
Longform (Köksal et al., 2024)	Wikipedia	4,483	100
ExpertQA (Malaviya et al., 2024)	Expert	1,741	100
Total	-	10,577	300

Table 1: Basic statistics of our seed datasets. We specify the number of questions in each split. We obtain q_{base} from these datasets constructed from various corpus.

q_{cov} (Section 3.3.3). Generated C^2 queries (*i.e.*, q_{base} and q_{cov}) are then utilized to select candidate outlines. For example, the answer to a q_{base} `"What is Generative AI?"` can contain diverse perspectives, including its latest trend, historical context, and application across different fields. Within available outlines that guide to satisfying answers, our goal is to obtain the outline that follows q_{cov} `"Tell me about its historical context"`. Therefore, within the range of QTREE, we parse candidate outlines for each C^2 query (Section 3.4.1), preceded by the evaluation for selecting the optimal outline (Section 3.4.2). The following subsections detail the procedural generation, and all used prompts are provided in Appendix E.

3.3 Preparing C^2 Queries (`Step 1`)

3.3.1 Base Query (q_{base}) Collection

We first collect q_{base} that requires long-form content composition to respond. Specifically, we employ two Wikipedia-based long-form question answering datasets – ASQA (Stelmakh et al., 2022) and Longform (Köksal et al., 2024), and one from expert domains – ExpertQA (Malaviya et al., 2024), as demonstrated in Table 1. For the test set, we sample 100 test queries for each dataset. By leveraging LLMs[2], we construct C^2 queries by combining these 10K q_{base} with corresponding q_{cov} (see Section 3.3.3). We slightly modify and filter a few q_{base} containing noises, described in Appendix A.

3.3.2 QTREE Construction

Prior to generating q_{cov}, we decompose q_{base} into diverse subqueries as a tree structure (*i.e.*, QTREE). The purpose of constructing QTREE for each q_{base} is to unfold the scope of information within parametric knowledge of LLMs. This structured graph also enables effective instruction generation (will be detailed in the next subsection) according to the hierarchy of abstractiveness. Subqueries in deeper depth present more specific subtopics. Table 6 in Appendix C illustrates an example of QTREE.

Quality Check In the query decomposition stage, we ensure that QTREE contains a predefined number of subqueries (*i.e.*, three) in each depth and width and does not overlap each other. This can be simply done by heuristically inspecting and comparing the structured output.

3.3.3 Coverage Query (q_{cov}) Generation

To remind, the role of q_{cov} is to specify certain subtopics to address (*i.e.*, include or exclude) within a broad range of information. Therefore, generating q_{cov} from QTREE requires selecting a specific viewpoint to cover. However, solely relying on LLMs' parametric knowledge does not guarantee the diversity of realistic situations. We therefore adopt the following two concepts to assist in generating q_{cov}.

- ***Background Subquery***: Understandably, asking for specific knowledge means that users are recognizing the knowledge itself. With this in consideration, we randomly select a single subquery from QTREE, which will be the knowledge users are aware of. We define this subquery as **background subquery**. The specificity of the background subquery differs according to the depth of the selected query.

- ***Intent Operation***: While considering a particular subject to ask, users may choose whether the content should be addressed within the responses. We conceptualize user intent through a binary operation (*i.e.*, INCLUSION, EXCLUSION), thereby facilitating the generation of q_{cov} that explicitly request the inclusion/exclusion of the subtopic on the background subquery.

In practice, we prompt LLM to generate q_{cov} by combining a randomly selected background subquery from QTREE with intent operation[3]. As

[2] We use `gpt-4-0125-preview` of OpenAI (OpenAI, 2023) with a temperature of 1.0, throughout this work.

[3] Although we use background subquery to generate q_{cov}

Intent Operation	q_{cov} Examples
INCLUSION	▶ Considering my eagerness to learn about educational analysis, include any thematic discussions by experts on the qualifications or contributions of the newly appointed UPSC member to the commission. ▶ Since I'm curious about the roots of the name, please explain where the name Jibril originated from. ▶ Given my interest in agriculture, include details about how different seasons can enhance or diminish the quality and quantity of tea produced in various regions.
EXCLUSION	▶ Ensure you omit any irrelevant details about Mary Poppins itself; I'm only interested in the birth date of the actress who played the bird lady. ▶ Since I already understand the elements required to prove theft, ensure to focus on the different classifications of theft in various legal systems without delving into the proof elements. ▶ Avoid diving into the biographies of other directors from the series; I'm only interested in the one who directed the initial movie.

Table 2: Example of generated q_{cov} according to intent operations (randomly sampled from the training set).

C^2 Query:
Describe the film The Woman Hunt. Since I'm already familiar with how audiences and critics received The Woman Hunt, please avoid discussing reviews or reception in your explanation.

Parsed Outline:

1. What is the plot of The Woman Hunt?
    1.1. What are the main events in The Woman Hunt?
        1.1.1. What initiates the conflict in The Woman Hunt?
        1.1.2. What is the climax of The Woman Hunt?

Table 3: Example of parsed outline. Example of corresponding QTREE is available at Table 6.

demonstrated in Table 2, combinations of background subquery and intent operation yield diverse q_{cov} for each q_{base}. Especially, requirements of q_{cov} with EXCLUSION operation are more complicated (*e.g.*, avoiding one topic but focusing on another topic) than INCLUSION. We analyze the performance difference according to intent operations in Appendix D. We sample five preliminary q_{cov} per each q_{base} and finally choose one if corresponding three candidate outlines are parsed correctly (which will be further described in Section 3.4.1).

3.4 Exploring Candidate Outlines & Evaluation (`Step 2`)

3.4.1 Parsing Outlines

In this stage, LLM sequentially extracts JSON-formatted candidate outlines from QTREE that satisfy instructions of C^2 queries[4]. Table 3 visualizes the example of a candidate outline, consisting of hierarchical subqueries (*i.e.*, plot - main event - conflict & climax) about *The Woman Hunt*. We extract three candidate outlines per each C^2 query.

Quality Check We fix the number of subqueries within each outline to four, guaranteeing that all subqueries are directly connected or neighboring within QTREE, as illustrated in Table 3. Additionally, we verify the JSON parsability of each outline and ensure that all subqueries do not overlap each other. For the efficient usage of API calls, we heuristically remove subqueries in leaf nodes if an outline contains more than four subqueries and include the outline as a candidate.

3.4.2 Evaluating Outline Quality

In order to rank three candidate outlines, we leverage LLM (`gpt-4-0125-preview`) to serve as a judge deciding whether the content on each candidate outline follows C^2 query. More precisely, we prompt the model to assign five-point Likert-scale

in this section, it is also used to construct baselines. Refer to Section 4.2.

[4] Our preliminary verification identifies that sequentially generating candidate outlines shows more diversity than temperature-based sampling. Refer to Appendix B for case studies.

scores with rationales, measuring how faithfully the outline aligns with the C^2 query. Since outlines are significantly shorter than long-form text while maintaining core contents (Kellogg, 1988), it is expected that evaluating outlines is more efficient and intuitive than directly evaluating long responses. These scored outlines are utilized as supervision and alignment pairs for training QPlanner, which will be described in further sessions.

3.5 Training QPlanner (Step 3)

To generalize with arbitrary C^2 queries, we train a 7B language model named QPlanner. We instruct QPlanner to sequentially generate QTree and select an outline, as we intend that QTree serves like an intermediate Chain-of-Thought (Wei et al., 2022) reasoning process. More technical details are described in Section 4.1.

4 Experiments

4.1 Training Details of QPlanner

We employ supervised fine-tuning (SFT) and alignment tuning for the training QPlanner. First, we train the Llama-2-7B-Chat model (Touvron et al., 2023) using 10K C^2 queries mapped with 31K candidate outline pairs, constructed through Section 3.4.1. This training phase allows the model to generate formatted outlines following C^2 queries (named **SFT-QPlanner** hereafter).

Then we further align the preferred outline by adopting a variant of direct preference optimization (DPO) (Rafailov et al., 2023). Following Tunstall et al. (2024) that show the possibility of distilling the preference of large teacher models into a targeted model, we utilize LLM evaluation scores previously acquired in Section 3.4.2 as reward signals for aligning QPlanner (named **DPO-QPlanner** hereafter). We regard the highest-scored outline as a positive (chosen) sample and the lowest-scored outline as a negative (rejected) sample. We skip samples whose highest and lowest scores are the same in the alignment stage.

The amount of the final training sample is 31,488 for SFT-QPlanner and 8,568 for DPO-QPlanner, respectively. Refer to Appendix F for further details, such as hyperparameters.

4.2 Baselines for Outline Comparison

Random Baseline Since the output of QPlanner accompanies QTree as an intermediate reasoning process, we can extract an arbitrary outline by leveraging this, regardless of the C^2 queries. Specifically, we select a random background subquery from QTree generated by SFT-QPlanner, then extend the branch to randomized directions (*i.e.*, upper depth, neighbor, or lower depth) until four subqueries are connected as a single outline. Intent operation is not considered in this random baseline.

	Mean (↑)	**SD** (↓)
Random Basline	2.57	1.44
SFT-QPlanner (31K)	2.79	1.40
DPO-SynNeg (31K + 8K align)	2.98	1.39
DPO-Combined (31K + 16K align)	3.01	1.36
DPO-QPlanner (Ours; 31K + 8K align)	**3.16**	**1.33**

Table 4: Mean and standard deviation (SD) for automatic outline evaluation (five-point Likert scale). DPO-QPlanner scores the highest mean score and the lowest SD, indicating robust improvement.

DPO-SynNeg To further explore the effectiveness of selected (*i.e.*, LLM-scored) negative samples in DPO-QPlanner, we prepare another DPO model trained with different types of negative samples. While negative samples of DPO-QPlanner are based on LLM scores, we can also heuristically synthesize negative samples with QTree, background subquery, and intent operation. This procedure is similar to generating random baseline, except for ensuring that synthesized outlines have the *opposite* intent operation to the original intent. For example, if the positive outline *includes* background subquery, the synthesized outline is designed to *exclude* that subquery by selecting another random background subquery within QTree. For the opposite situation, the synthesized outline must contain background subquery of the positive outline. On the 8K DPO-QPlanner training set, we maintain the positive samples and replace negative samples with synthetically generated outlines.

DPO-Combined We also measure the performance of combining negative samples in DPO-QPlanner and DPO-SynNeg. That is, the number of training samples is doubled.

5 Results

5.1 Automatic Outline Evaluation

We prompt LLM (`gpt-4-0125-preview`) to score generated outlines in the test set. We use scoring rubric in Table 9 in Appendix G.1.

5.1.1 Mean Score Comparison

Table 4 shows our test result with a five-point Likert scale. We score outlines generated by each trained model[5], focusing on whether the content of outlines follows given C^2 queries (as mentioned in Section 3.4.2).

We find that the random baseline shows the lowest mean score (2.57) with the highest standard deviation (1.44) on our test set. While SFT-QPLANNER shows a higher score than the random baseline, we find that DPO-QPLANNER significantly improves the score (3.16) with the lowest standard deviation (1.33). It implies that we can **leverage LLM-generated scores as reward signals in query outlining** when constructing positive-negative pairs, even in the absence of explicit and gold reward criteria for their construction (Ma et al., 2023; Yoon et al., 2024).

On the comparison with DPO-QPLANNER and DPO-SYNNEG, we observe that **negative samples of DPO-QPLANNER are notably more effective** than the other one, since the only difference between them is the type of negative samples. We conjecture that constructing "hard" negatives[6] (*i.e.*, less scored subtrees with the "same" intent) is an important factor to align with C^2 queries, sharing insights with different studies on hard negatives (Rosset et al., 2023; Scarlatos et al., 2024). Regarding DPO-COMBINED, the performance becomes worse than DPO-QPLANNER despite the doubled amount of alignment pairs, implying the importance of selective negative samples.

5.1.2 Pairwise Comparison

For comprehensive evaluation, we also compare pairwise scores among models. As illustrated in Figure 3, **generated outlines of DPO-QPLANNER are more preferred than all other baselines on the same C^2 query**, which is aligned with the atomic scoring result in Table 4. Given the fact that exhaustively devising preferred outlines for C^2 queries is labor-intensive, our QPLANNER is an effective solution for exploring and creating preferred outlines for long-form responses.

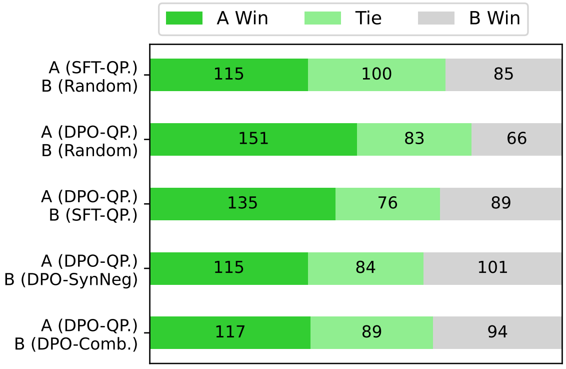


Figure 3: Pairwise comparison for each C^2 query in automatic outline evaluation.

5.2 Human Outline Evaluation

We conduct a human study to identify the effectiveness of QPLANNER. We describe detailed experimental setup in Appendix G, such as demographics and payment.

We let participants read and rate two outlines generated from SFT- and DPO-QPLANNER for randomly selected 100 C^2 queries from the test set using the same five-point Likert scale criteria as the automatic evaluation. Each evaluator rates from 4 to 10 outlines in random order (to avoid position bias), and each outline has at least 6 evaluators (Average: 6.55, Max: 15). We intend to have as many evaluators as possible to rate individual outlines to gather a collective rating for each set. This is because even with simple outlines, judging outlines with unfamiliar topics is a highly intellectual and unavoidably subjective task.

Consequently, we find **significant positive correlations between human-rated scores and LLM-rated scores** – both for SFT (Pearson's r = 0.51, p-value $<$ 0.001) and DPO-QPLANNER (Pearson's r = 0.39, p-value $<$ 0.001), which indicates positive relationships with large and medium strength, respectively. Moreover, DPO-QPLANNER receives higher human scores (Mean= 3.29, Std=0.81) than SFT (Mean= 3.03, Std=0.78)[7].

[5] A few cases return an outline with 3 or 5 queries, which is not an ideal number of the output (*i.e.*, 4), but we do not filter them in our evaluation.

[6] We define LLM-generated negative samples as "hard" when compared to synthetic negative samples whose intent operations are explicitly opposite to positive outlines.

[7] This trend is supported even when we regress scores on model version (SFT or DPO-QPLANNER) and the total length of outlines in characters, with outline id as a fixed effect (Model:b=0.27, p-value=0.01; Outline length: b=0.01, p-value=0.30). That is, the length of outlines is not predictive of scores.

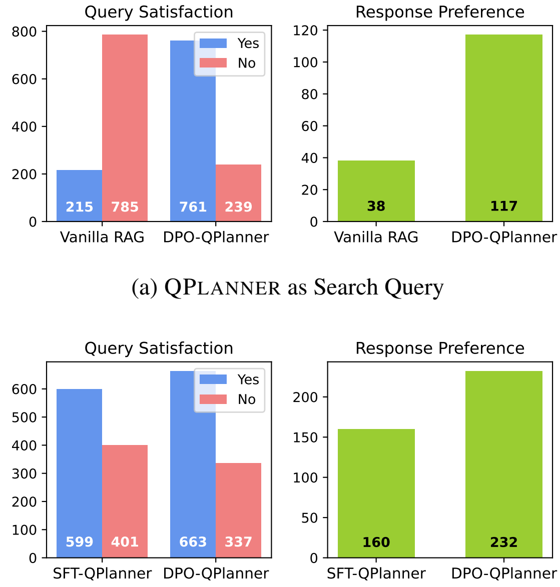
We report that evaluating highly subjective tasks may introduce varied ratings among human evaluators despite assigning a large number of evaluators to derive the majority opinion (Krippendorff's α: SFT-QPlanner = 0.22; DPO-QPlanner = 0.23), as observed in other studies (Rottger et al., 2022; Abercrombie et al., 2023).

5.3 Human RAG Evaluation

For long-form response evaluation, we do not automatically measure due to the lack of reliability in long-form text evaluation (Xu et al., 2023). Instead, we recruit another participant to validate the effectiveness of QPlanner on RAG downstream tasks[8]. 100 RAG responses from the test set are sampled for evaluation, and ten evaluators are assigned for each response. Following insights from Kim et al. (2024b) where the writing format of model responses affects human preferences, we fix the response format with Markdown to compare responses by focusing only on their content. In addition, we prompt LLM[9] to generate responses by strictly relying on given evidence to prevent LLM from arbitrarily responding with its parametric knowledge. We assume web search scenarios for the RAG setup, providing detailed information in Appendix G.2.1.

Regarding evaluation criteria, we first instruct participants to judge whether generated responses follow requirements of C^2 queries or not (`Query Satisfaction` in Figure 4). We guide them to annotate "Yes" if responses at least partially address topics within C^2 queries. For response pairs annotated as "Yes" in both models, participants select their preferred response (`Response Preference` in Figure 4).

QPlanner as Better Search Query We verify whether subqueries within outlines can help search relevant documents. We compare responses of vanilla RAG with those of DPO-QPlanner using the exactly same prompt. That is, subqueries of DPO-QPlanner only affect the search result. As illustrated in Figure 4a, we observe that the conventional RAG pipeline does not properly retrieve relevant evidence for answering C^2 queries, whose requirements are far more complicated than normal queries. Furthermore, among responses that satisfy the requirements of C^2 queries, responses of DPO-QPlanner are mostly preferred.



(a) QPlanner as Search Query

(b) QPlanner as Content Draft

Figure 4: Human evaluation results.

QPlanner as Better Content Draft We further investigate whether better outlines lead to better responses. In this setup, we compare responses of SFT-QPlanner and DPO-QPlanner. The exactly same prompt is used for this comparison, and subqueries within outlines are included in the prompt for composing responses and retrieving documents. Results in Figure 4b indicate that further aligning QPlanner with preference can provide preferred outlines, while SFT-QPlanner also shows a similar tendency with DPO-QPlanner.

Since the outcome of both studies is binary (*i.e.*, satisfactory or not), and each human evaluator judges both responses for the same query (within-subjects design), we conduct two McNemar's tests (McNemar, 1947) to examine whether the differences we find are statistically significant. The contingency tables used for the tests can be found in Table 10. The results confirm that DPO-QPlanner significantly outperforms Vanilla RAG (test-statistics = 60, p-value < 0.001) and SFT-QPlanner (test-statistics = 207, p-value = 0.004).

6 Conclusion

In this work, we suppose complicated user scenarios asking for a constrained range of a specific topic, called coverage-conditioned (C^2) query sce-

[8] Details such as recruitment, instructions, and compensation are described in Appendix G.2

[9] `gpt-4-0125-preview` is used.

narios. To simulate C^2 scenarios and a controlled environment for creating and evaluating query outlines (*RQ1*), we construct QTREE, hierarchical sets of subqueries representing diverse perspectives of the original query. Playing a role as boundaries for available outlines, QTREE allows systematic comparison of diverse outlines. Subsequently, we train QPLANNER which extracts customized outlines from QTREE for C^2 queries. Regarding our *RQ2*, our findings based on automatic and human evaluation show that (1) preference-aligned QPLANNER can generate better outlines, (2) outlines enable improved document search, and (3) better outlines lead to preferred responses. We believe our work shows the possibility of QTREE as a testbed for exploring effective pre-writing strategies to deal with complicated queries.

Limitations

We discuss the current limitations of our work. First, our graphical representation of subquery nodes adheres to canonical tree structures, with each node connected to three child nodes, but it can be adjusted (*i.e.*, composing more or less subqueries) according to tasks or domains. For example, in a complex domain like medical diagnosis, a larger number of subqueries might be necessary to cover various symptoms, possible conditions, diagnostic tests, and treatment options. In contrast, for a straightforward factual query in a domain like mathematics, fewer subqueries might be sufficient to reach a comprehensive answer. Identifying this optimal number still remains an open question and represents a promising direction for future investigation. We believe that our experimental setup serves as an initial testbed for validating these research questions.

It should also be noted that the contents of retrieved documents in our RAG setup can affect the detailed factual consistency of final responses. Although we set the same search configuration among all methodologies, additional fact verification of documents and responses is still needed for practical applications.

While demonstrating significant performance gains both in automatic and human judgements, we find that state-of-the-art LLMs still have difficulty generating long-form responses that handle detailed coverage of C^2 queries. This is presumably due to the complexness of C^2 queries, and it arises the importance of constructing meticulous benchmarks evaluating long-form responses to complicated queries. This will be another direction of the future work. Lastly, we would like to mention that our five-point scoring schema can be further improved by considering multiple aspects with a fine-grained score rubric.

Ethical Considerations

Since our QTREE is generated with benchmarks based on Wikipedia and domain experts, we do not filter sensitive or unsafe contents throughout our studies. For the practical application in the future, deliberate content selection will be required for the safety. In addition, we explicitly share our experimental setup of human studies for transparency in Appendix G.

References

Gavin Abercrombie, Verena Rieser, and Dirk Hovy. 2023. Consistency is key: Disentangling label variation in natural language processing with intra-annotator agreement. *Preprint*, arXiv:2301.10684.

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. 2018. Ms marco: A human generated machine reading comprehension dataset. *Preprint*, arXiv:1611.09268.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53.

Pradeep Dasigi, Kyle Lo, Iz Beltagy, Arman Cohan, Noah A. Smith, and Matt Gardner. 2021. A dataset of information-seeking questions and answers anchored in research papers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4599–4610, Online. Association for Computational Linguistics.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. ELI5: Long form question answering. In *Proceedings of* *the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3558–3567, Florence, Italy. Association for Computational Linguistics.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898, Melbourne, Australia. Association for Computational Linguistics.

Sebastian Gehrmann, Elizabeth Clark, and Thibault Sellam. 2023. Repairing the cracked foundation: A survey of obstacles in evaluation practices for generated text. *J. Artif. Int. Res.*, 77.

Google. 2024. Gemini: A family of highly capable multimodal models. *Preprint*, arXiv:2312.11805.

Rolf Jagerman, Honglei Zhuang, Zhen Qin, Xuanhui Wang, and Michael Bendersky. 2023. Query expansion by prompting large language models. *Preprint*, arXiv:2305.03653.

Ronald T. Kellogg. 1988. Attentional overload and writing performance: effects of rough draft and outline strategies. *Journal of Experimental Psychology: Learning, Memory and Cognition*, 14:355–365.

Tushar Khot, Harsh Trivedi, Matthew Finlayson, Yao Fu, Kyle Richardson, Peter Clark, and Ashish Sabharwal. 2023. Decomposed prompting: A modular approach for solving complex tasks. In *The Eleventh International Conference on Learning Representations*.

Seungone Kim, Jamin Shin, Yejin Cho, Joel Jang, Shayne Longpre, Hwaran Lee, Sangdoo Yun, Seongjin Shin, Sungdong Kim, James Thorne, and Minjoon Seo. 2024a. Prometheus: Inducing fine-grained evaluation capability in language models. In *The Twelfth International Conference on Learning Representations*.

Yoonsu Kim, Jueon Lee, Seoyoung Kim, Jaehyuk Park, and Juho Kim. 2024b. Understanding users' dissatisfaction with chatgpt responses: Types, resolving tactics, and the effect of knowledge level. In *Proceedings of the 29th International Conference on Intelligent User Interfaces*, IUI '24, page 385–404, New York, NY, USA. Association for Computing Machinery.

Kalpesh Krishna, Erin Bransom, Bailey Kuehl, Mohit Iyyer, Pradeep Dasigi, Arman Cohan, and Kyle Lo. 2023. LongEval: Guidelines for human evaluation of faithfulness in long-form summarization. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1650–1669, Dubrovnik, Croatia. Association for Computational Linguistics.

Kalpesh Krishna, Aurko Roy, and Mohit Iyyer. 2021. Hurdles to progress in long-form question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4940–4957, Online. Association for Computational Linguistics.

Abdullatif Köksal, Timo Schick, Anna Korhonen, and Hinrich Schütze. 2024. Longform: Effective instruction tuning with reverse instructions. *Preprint*, arXiv:2304.08460.

Yoonjoo Lee, Kyungjae Lee, Sunghyun Park, Dasol Hwang, Jaehyeon Kim, Hong-In Lee, and Moontae Lee. 2023. QASA: Advanced question answering on scientific articles. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 19036–19052. PMLR.

Yukyung Lee, Soonwon Ka, Bokyung Son, Pilsung Kang, and Jaewook Kang. 2024. Navigating the path of writing: Outline-guided text generation with large language models. *Preprint*, arXiv:2404.13919.

E. Levin, R. Pieraccini, and W. Eckert. 2000. A stochastic model of human-machine interaction for learning dialog strategies. *IEEE Transactions on Speech and Audio Processing*, 8(1):11–23.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, Benjamin Newman, Binhang Yuan, Bobby Yan, Ce Zhang, Christian Alexander Cosgrove, Christopher D Manning, Christopher Re, Diana Acosta-Navas, Drew Arad Hudson, Eric Zelikman, Esin Durmus, Faisal Ladhak, Frieda Rong, Hongyu Ren, Huaxiu Yao, Jue WANG, Keshav Santhanam, Laurel Orr, Lucia Zheng, Mert Yuksekgonul, Mirac Suzgun, Nathan Kim, Neel Guha, Niladri S. Chatterji, Omar Khattab, Peter Henderson, Qian Huang, Ryan Andrew Chi, Sang Michael Xie, Shibani Santurkar, Surya Ganguli, Tatsunori Hashimoto, Thomas Icard, Tianyi Zhang, Vishrav Chaudhary, William Wang, Xuechen Li, Yifan Mai, Yuhui Zhang, and Yuta Koreeda. 2023. Holistic evaluation of language models. *Transactions on Machine Learning Research*. Featured Certification, Expert Certification.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: NLG evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522, Singapore. Association for Computational Linguistics.

Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. 2023. Query rewriting in retrieval-augmented large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5303–5315, Singapore. Association for Computational Linguistics.

Chaitanya Malaviya, Subin Lee, Sihao Chen, Elizabeth Sieber, Mark Yatskar, and Dan Roth. 2024. ExpertQA: Expert-curated questions and attributed answers. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3025–3045, Mexico City, Mexico. Association for Computational Linguistics.

Quinn McNemar. 1947. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12076–12100, Singapore. Association for Computational Linguistics.

OpenAI. 2023. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Corby Rosset, Guoqing Zheng, Victor Dibia, Ahmed Awadallah, and Paul Bennett. 2023. Axiomatic preference modeling for longform question answering. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 11445–11475, Singapore. Association for Computational Linguistics.

Paul Rottger, Bertie Vidgen, Dirk Hovy, and Janet Pierrehumbert. 2022. Two contrasting data annotation paradigms for subjective NLP tasks. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 175–190, Seattle, United States. Association for Computational Linguistics.

Alexander Scarlatos, Digory Smith, Simon Woodhead, and Andrew Lan. 2024. Improving the validity of automatically generated feedback via reinforcement learning. *Preprint*, arXiv:2403.01304.

Yijia Shao, Yucheng Jiang, Theodore Kanell, Peter Xu, Omar Khattab, and Monica Lam. 2024. Assisting in writing Wikipedia-like articles from scratch with large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6252–6278, Mexico City, Mexico. Association for Computational Linguistics.

Toby Shevlane, Sebastian Farquhar, Ben Garfinkel, Mary Phuong, Jess Whittlestone, Jade Leung, Daniel Kokotajlo, Nahema Marchal, Markus Anderljung, Noam Kolt, Lewis Ho, Divya Siddarth, Shahar Avin, Will Hawkins, Been Kim, Iason Gabriel, Vijay Bolina, Jack Clark, Yoshua Bengio, Paul Christiano, and Allan Dafoe. 2023. Model evaluation for extreme risks. *Preprint*, arXiv:2305.15324.

Ivan Stelmakh, Yi Luan, Bhuwan Dhingra, and Ming-Wei Chang. 2022. ASQA: Factoid questions meet long-form answers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8273–8288, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Xiaofei Sun, Zijun Sun, Yuxian Meng, Jiwei Li, and Chun Fan. 2022. Summarize, outline, and elaborate: Long-text generation via hierarchical supervision from extractive summaries. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 6392–6402, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Lewis Tunstall, Edward Emanuel Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro Von Werra, Clémentine Fourrier, Nathan Habib, Nathan Sarrazin, Omar Sanseviero, Alexander M Rush, and Thomas Wolf. 2024. Zephyr: Direct distillation of LM alignment. In *First Conference on Language Modeling*.

Liang Wang, Nan Yang, and Furu Wei. 2023. Query2doc: Query expansion with large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9414–9423, Singapore. Association for Computational Linguistics.

Tiannan Wang, Jiamin Chen, Qingrui Jia, Shuai Wang, Ruoyu Fang, Huilin Wang, Zhaowei Gao, Chunzhao Xie, Chuou Xu, Jihong Dai, Yibin Liu, Jialong Wu, Shengwei Ding, Long Li, Zhiwei Huang, Xinle Deng, Teng Yu, Gangan Ma, Han Xiao, Zixin Chen, Danjun Xiang, Yunxia Wang, Yuanyuan Zhu, Yi Xiao, Jing Wang, Yiru Wang, Siran Ding, Jiayang Huang, Jiayi Xu, Yilihamu Tayier, Zhenyu Hu, Yuan Gao, Chengfeng Zheng, Yueshu Ye, Yihang Li, Lei Wan, Xinyue Jiang, Yujie Wang, Siyu Cheng, Zhule Song, Xiangru Tang, Xiaohua Xu, Ningyu Zhang, Huajun Chen, Yuchen Eleanor Jiang, and Wangchunshu Zhou. 2024. Weaver: Foundation models for creative writing. *Preprint*, arXiv:2401.17268.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc.

Fangyuan Xu, Junyi Jessy Li, and Eunsol Choi. 2022. How do we answer complex questions: Discourse structure of long-form answers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3556–3572, Dublin, Ireland. Association for Computational Linguistics.

Fangyuan Xu, Yixiao Song, Mohit Iyyer, and Eunsol Choi. 2023. A critical evaluation of evaluations for long-form question answering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3225–3245, Toronto, Canada. Association for Computational Linguistics.

Kevin Yang, Dan Klein, Nanyun Peng, and Yuandong Tian. 2023. DOC: Improving long story coherence with detailed outline control. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3378–3465, Toronto, Canada. Association for Computational Linguistics.

Kevin Yang, Yuandong Tian, Nanyun Peng, and Dan Klein. 2022. Re3: Generating longer stories with recursive reprompting and revision. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4393–4479, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Seonghyeon Ye, Doyoung Kim, Sungdong Kim, Hyeonbin Hwang, Seungone Kim, Yongrae Jo, James Thorne, Juho Kim, and Minjoon Seo. 2024. FLASK: Fine-grained language model evaluation based on alignment skill sets. In *The Twelfth International Conference on Learning Representations*.

Chanwoong Yoon, Gangwoo Kim, Byeongguk Jeon, Sungdong Kim, Yohan Jo, and Jaewoo Kang. 2024. Ask optimal questions: Aligning large language models with retriever's preference in conversational search. *Preprint*, arXiv:2402.11827.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. BARTScore: Evaluating generated text as text generation. In *Advances in Neural Information Processing Systems*.

A Base Query Modification

For the ASQA[10] (Stelmakh et al., 2022) and ExpertQA[11] (Malaviya et al., 2024) dataset, we do not modify the base query. For the Longform[12] (Köksal et al., 2024) dataset, as there are additional format-related instructions concatenated with the base query (*e.g.*, *Respond in 3 sentences.*), we eliminate them by using regular expressions. Moreover, we find that Longform dataset contains noisy queries (*e.g.*, *This does not provide enough information for an answer to be provided.*), which are unfiltered artifacts generated by large language models. In this case, we manually filter similar expressions.

B Outline Sampling Comparison

To identify the effectiveness of sequentially generating candidate outlines at once, we generate candidate outlines using temperature sampling. As shown in Table 5, sequential generation meaningfully diversifies candidate outlines than temperature sampling. Similar results are obtained when we increase the sampling temperature.

[10] Apache 2.0 License
[11] MIT License
[12] MIT License

C^2 Query:
When was the museum of islamic art built? Since I'm already familiar with who the lead architect of the Museum of Islamic Art was and their design vision, focus on providing the construction date without delving into architectural design specifics.

Parsed Outline (Sampling Temperature 1.0):

3.2. Which construction company undertook the building of the Museum of Islamic Art, and in which year did they commence work?
3.2.1. What firm was responsible for the construction of the Museum of Islamic Art, and when did they start ?
3.2.2. Can you identify the construction firm that built the Museum of Islamic Art and the year construction began?
3.2.3. Who constructed the Museum of Islamic Art, and in what year did construction efforts begin?

3.3. What were the financial and logistical challenges in constructing the Museum of Islamic Art?
3.3.1. What budgetary constraints affected the construction of the Museum of Islamic Art?
3.3.2. What logistical hurdles were encountered during the building of the Museum of Islamic Art?
3.3.3. In constructing the Museum of Islamic Art, what were the major financial and logistical issues

3. Who contributed to the design and construction of the Museum of Islamic Art?
3.2. Which construction company undertook the building of the Museum of Islamic Art, and in which year did they commence work?
3.2.1. What firm was responsible for the construction of the Museum of Islamic Art, and when did they start ?
3.2.2. Can you identify the construction firm that built the Museum of Islamic Art and the year construction began?

Candidate Outlines (Sequential Generation):

2. When did the Museum of Islamic Art open to the public ?
2.1. What is the official opening date of the Museum of Islamic Art?
2.1.1. On which date was the Museum of Islamic Art inaugurated ?
2.1.2. When was the grand opening of the Museum of Islamic Art?

1. What is the construction date of the Museum of Islamic Art?
1.2. Who was the architect behind the Museum of Islamic Art design, and in what year did the construction take place under their guidance?
1.2.1. What is the name of the architect who designed the Museum of Islamic Art, and during which year was their design realized ?
1.2.3. In what year did construction of the Museum of Islamic Art occur under the designated architect 's design?

3. Who contributed to the design and construction of the Museum of Islamic Art?
3.2. Which construction company undertook the building of the Museum of Islamic Art, and in which year did they commence work?
3.2.1. What firm was responsible for the construction of the Museum of Islamic Art, and when did they start ?
3.2.3. Who constructed the Museum of Islamic Art, and in what year did construction efforts begin?

Table 5: Comparison of temperature sampling and sequential generation of candidate outlines.

C Example of QTree

C^2 Query:
Describe the film The Woman Hunt. Since I'm already familiar with how audiences and critics received The Woman Hunt, please avoid discussing reviews or reception in your explanation.

QTree:

1. What is the plot of The Woman Hunt?
1.1. What are the main events in The Woman Hunt?
1.1.1. What initiates the conflict in The Woman Hunt?
1.1.2. What is the climax of The Woman Hunt?
1.1.3. How does The Woman Hunt end?
1.2. Who are the main characters in The Woman Hunt?
1.2.1. Who is the protagonist of The Woman Hunt?
1.2.2. Who is the antagonist in The Woman Hunt?
1.2.3. What supporting characters play crucial roles in The Woman Hunt?
1.3. What themes are explored in The Woman Hunt?
1.3.1. What is the primary theme of The Woman Hunt?
1.3.2. How does The Woman Hunt explore gender dynamics?
1.3.3. What messages does The Woman Hunt convey about survival?
2. Who directed The Woman Hunt?
2.1. What is the directorial style of The Woman Hunt?
2.1.1. How does the director use camera angles in The Woman Hunt?
2.1.2. What unique directorial choices are made in The Woman Hunt?
2.1.3. How does the pace affect the narrative in The Woman Hunt?
2.2. What other films has the director of The Woman Hunt made?
2.2.1. What are the most popular films by The Woman Hunt's director?
2.2.2. How do other films by the director compare to The Woman Hunt?
2.2.3. What recurring themes appear in the director ' s filmography?
2.3. How has the director ' s background influenced The Woman Hunt?
2.3.1. What aspects of the director ' s personal life reflect in The Woman Hunt?
2.3.2. How does the director ' s cultural background inform The Woman Hunt?
2.3.3. What previous experiences of the director shaped The Woman Hunt?
3. How was The Woman Hunt received by audiences and critics ?
3.1. What are the critical reviews of The Woman Hunt?
3.1.1. How do film critics analyze The Woman Hunt?
3.1.2. What are the predominant critiques of The Woman Hunt?
3.1.3. Are there any notable defenses of The Woman Hunt's thematic choices?
3.2. What is the audience ' s reaction to The Woman Hunt?
3.2.1. How do audience perspectives on The Woman Hunt vary?
3.2.2. What aspects of The Woman Hunt resonate most with audiences?
3.2.3. What fan opinions of The Woman Hunt diverge from critical reviews?
3.3. Has The Woman Hunt won any awards or recognition?
3.3.1. What awards or nominations has The Woman Hunt received?
3.3.2. How does The Woman Hunt rank among other films of its genre?
3.3.3. Are there any film festivals where The Woman Hunt was highlighted?

Table 6: Example of QTree generated by the process described in Section 3.3.2.

D Analysis on Intent Operations

	SFT-QPLANNER		DPO-QPLANNER	
	INCLUSION	EXCLUSION	INCLUSION	EXCLUSION
Mean	2.85	2.74	3.22	3.10
SD	1.23	1.55	1.15	1.47

Table 7: Mean and standard deviation (SD) according to the intent operation in automatic outline evaluation.

We decompose the result of Table 4 according to intent operations in Table 7, focusing on SFT-QPLANNER and DPO-QPLANNER scores. We discover that C^2 queries based on EXCLUSION score lower than those on the intent of INCLUSION. This result aligns with our assumption in Section 3.3.3, where q_{cov} with EXCLUSION operation require more complicated selection of desirable outline.

E Used Prompts

We use the following prompts in our work.

Decomposing Base Query into QTree

~~~
You are a subquery graph generator.
Generate a numbered tree graph of given question with three-depth hierarchies. You also need to follow the instructions below:

All keys and values should be in the form of question. Each node in the first depth should provide unique viewpoint interpreting the question. The number of siblings and leaves should be three.
Output format should be a dictionary format. For example, the output should be expressed as
```json {1. node: {1.1. node: [1.1.1. node, 1.1.2. node, 1.1.3. node], 1.2. node: ...}, 2. node: {...}, ... ```
All subqueries should be concise, independent and self-contained from other context so that it can be used as a search query. All subqueries (keys and values, and each other) should not be overlapped.

Question: {{base_query}}
Subquery Graph:
~~~

Generating Coverage Queries Using Background Subquery and Intent Operation

You are an instruction generator.
Your role is to generate five self-contained instructions that relate subquestion with question.

The following is guidelines you must follow:

{{The user wants to know about the answer specifically related to the given subquestion. Therefore, generated instructions MUST ask to include topics of the subquestion.}} **OR** {{The user already knows about the answer to the given subquestion. Therefore, generated instructions MUST require to exclude topics of the subquestion.}}
Each instruction should be clear and concise. Moreover, please use various expressions for including/excluding specific topics. Each instruction should not contract with Question. Make sure to generate instructions that can be naturally connected after Question. Instructions should be written from a first person's viewpoint. For example, the instruction can start as 'Since I know/do not know about ..., you need to ...' Five instructions should be easily separated by "&&", without any numbering or bullet points.

Question: {{base_query}}
Subquestion: {{background_query}}
Instructions:

Parsing Outlines from QTree

You are a subtree parser.
Your role is to extract three json-formatted subtrees based on a full question tree, which is generated based on the root question "{{base_query}}"
Here is guidelines you must follow:

You should parse subtrees maintaining the original tree structure. Keys sharing the same parent node are allowed to parse, but completely separated keys are not allowed.
The first subtree should follow both 'Rule' and 'Instruction'. On the other hand, the second and the third subtree should follow 'Rule', but not necessarily 'Instruction'. The total number of nodes in each subtree should be 4. Subtrees should be separated with '&&' for easy parsing.

Tree: {{qtree}}
Rule: {{Include/Exclude the node ### in the subtree.}}
Instruction: {{coverage_query}}
Subtree:

Scoring Candidate Outlines Using Large Language Model

Your role is to evaluate the quality of subqueries generated from given query.

Query: {{coverage_conditioned_query}}
Subqueries:
{{list_of_search_queries}}

* Evaluation aspects
Instruction following: How accurately do the subqueries adhere to and execute the specific instructions provided?

* Scale
You should evaluate each aspect as a five-scale score metrics. Score 5 means that subqueries are highly following the evaluation aspects.

* Answer format
You should return a JSON-formatted response as demonstrated in the example.
Example: ```json {"rationale: "YOUR_RATIONALE_STRING", "score": YOUR_SCORE_INTEGER}```

Lastly, do not generate any other statements except for requested format.

QPlanner Input

Create a three-level deep query graph that expands on the knowledge related to the provided question. Next, identify and extract a subsection of this graph that most effectively answers the question, ensuring this subsection retains a tree-like structure and includes four distinct nodes.
Question: {{coverage_conditioned_query}}

Generating Response with Retrieved Documents (QPlanner for Search Query)

As an experienced writing assistant, your task is to generate an answer to the given query. Structure your answer in Markdown, employing elements like headings, subheadings, bullet points, and bold text for clarity and organization. Your response should strictly rely on the given evidence, so do not respond with information that is not given. In that case, just mention there is no evidence for that query.

QUERY: {{coverage_conditioned_query}}

EVIDENCE:

TITLE: {{title_1}}
CONTENT: {{content_1}}
...

RESPONSE:

Generating Response with Retrieved Documents (Qplanner for Search Query & Content Draft)

As an experienced writing assistant, your task is to generate an answer to the given query. Structure your answer in Markdown, employing elements like headings, subheadings, bullet points, and bold text for clarity and organization. Your response should strictly rely on the given evidence, so do not respond with information that is not given. In that case, just mention there is no evidence for that query. Also, there are subqueries that help answer the main query. You can use subtopics of subqueries as headers and the corresponding evidence as contents if they are useful to answer the main query.

QUERIES:
- Main Query: {{coverage_conditioned_query}}
- Subquery 1: {{query_1}}
- Subquery 2: {{query_2}}
- Subquery 3: {{query_3}}
- Subquery 4: {{query_4}}

EVIDENCE:

TITLE: {{title_1}}
CONTENT: {{content_1}}
SOURCE: {{query_used_to_search_this_document}}
...

RESPONSE:

F Training Details

Hyperparameter	SFT	DPO
Epoch	1	1
Batch Size Per Device	14	8
Learning Rate (LR)	2e-5	5e-7
LR Schedule	Cosine	Cosine
Warmup Ratio	0.1	0.1
Gradient Accumulation Step	1	2
Beta	-	0.01
# of Samples	31,488	8,568

Table 8: Hyperparameters for training QPLANNER. A few noisy samples are filtered in advance at SFT stage.

We utilize publicly available software[13] in our finetuning and alignment stage. We train each stage one epoch using 16 NVIDIA A100 GPUs (40GB of memory). Table 8 indicates detailed hyperparameters for each stage.

G Additional Information on Human Evaluation

We conduct two human evaluation studies separately (*i.e.*, outline and RAG evaluation). For both studies, crowdworkers are recruited from Prolific[14]. At the beginning of the evaluation, workers are informed what task they are expected to do, there are no foreseeable benefits and risks, their participation is voluntary, and they can leave if they want (see Figure 5).

G.1 Outline Evaluation

Score Rubric
1: The sub−questions/responses entirely disregards the instructions , providing content unrelated to the instruction .
2: The sub−questions/responses show a superficial attempt to follow instructions but significantly strays from the intended task , missing key objectives .
3: The sub−questions/responses generally adheres to the instructions but overlooks certain details or nuances, achieving only a partial match with the instruction .
4: The sub−questions/responses is closely aligned with the instructions , exhibiting minor deviations that slightly affect the completeness of the execution .
5: The sub−questions/responses exhibits impeccable adherence to the instructions , capturing all nuances and completing the task as specified .

Table 9: Score rubric for evaluating subqueries and responses in human evaluation.

A total of 127 crowd workers participate in the evaluation (Gender: 68 men, 57 women, and 2 non-binary; Age: Mean=28.6 yrs, SD=7.9 yrs, Min=18 yrs, Max=63 yrs; Ethnicity: White: 69, Black: 44, Mixed: 11, and Asian:2; Country of residence: South Africa: 52 (41.27%), Portugal: 20 (15.87%), Poland: 10 (7.94%), United Kingdom: 5 (3.97%), Mexico: 5 (3.97%), and 19 other countries; Highest education level completed: A majority of the evaluators hold at least a Bachelor's degree (n=83, 65.87%)).

[13] https://github.com/huggingface/alignment-handbook

[14] https://www.prolific.com/

Individual crowd workers evaluate different numbers of instances depending on their availability. They are compensated 9 GBP/hour for their work. We paid 606.41 GBP in total.

They score each sample by following the rubric in Table 9. We engage evaluators by asking them to write at least 30 characters to describe their rationale for preference, which also helps evaluators take the rating more seriously and derive more rational and accurate ratings. We also provide an evaluation session with no more than evaluating 5 outlines considering the human attention span. If evaluators want to continue participating, they have to sign up for another evaluation session, which ensures they have a break for recharging themselves. Lastly, we utilize Prolific's offering that automatically rejects work that takes too long or too short, above or below two standard deviations of the average completion time.

G.2 RAG Response Evaluation

Search Query Evaluation			**Content Draft Evaluation**		
	DPO-QPlanner			**DPO-QPlanner**	
Vanilla RAG	Unsatisfactory	Satisfactory	**SFT-QPlanner**	Unsatisfactory	Satisfactory
Unsatisfactory	179	606	Unsatisfactory	130	271
Satisfactory	60	155	Satisfactory	207	392

Note: The sum of the counts in the contingency tables is 1000 (100 queries evaluated by 10 workers) for each evaluation, respectively.

Table 10: Contingency tables for human study in response evaluation

A total of 63 crowd workers participate in the evaluation (Gender: 33 men and 30 women; Age: Mean=28.08 yrs, SD=9.01 yrs, Min=19 yrs, Max=68 yrs; Ethnicity: White: 26, Black: 22, Mixed: 10, and Asian: 5; Country of residence: South Africa: 20 (15.87%), Portugal: 10 (7.94%), Mexico: 8 (6.35%), Poland: 6 (4.76%), Canada: 4 (3.17%), and 11 other countries; Highest education level completed: A majority of the evaluators hold at least a Bachelor's degree (n=49, 77.78%)). Individual crowd workers evaluate different numbers of instances depending on their availability. They are compensated 9 GBP/hour for their work. We pay 878.58 GBP in total.

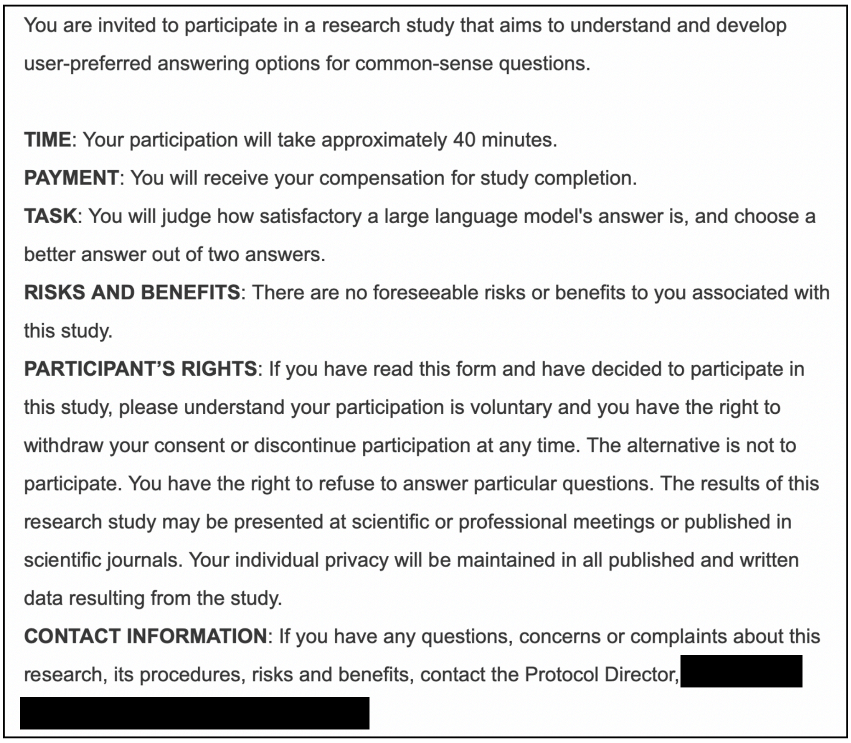
We offer ten single sessions for evaluation (5 sessions for Vanilla RAG vs. DPO-QPlanner and 5 sessions for SFT-QPlanner vs. DPO-QPlanner). Each session has ten evaluators. If wanted, evaluators can participate in more than one session; thirteen out of 63 evaluated multiple sessions. A session takes 20 to 40 minutes. In a session, evaluators are first provided with a short tutorial with evaluation guidelines and examples. Then, they evaluate a pair of responses that answer the same query for twenty queries. They are told that all of the queries are formatted as `[question]` (q_{base}) + `[instruction]` (q_{cov}) and asked to mark a response "Satisfactory" if the response satisfies any of the two evaluation items: (1) the response indeed answers the question `[question]` with evidence or partial evidence, which includes "there is no evidence but here is useful information," and (2) the response follows the instruction, `[instruction]`, and mark it "Unsatisfactory" otherwise. If all of the two responses to the same query are rated "Satisfactory," they were asked to choose which one was a better answer (155 cases in Search Query Evaluation and 392 cases in Response Outline Evaluation; refer to Table 10). We describe a better answer would have more items described above satisfied, or it would be better at following the instruction.

G.2.1 Document Retrieval

For simulating RAG pipeline, we utilize DuckDuckgo[15] to search relevant documents. To balance the number of documents, top-10 documents are retrieved in Vanilla RAG, and top-2 documents are retrieved for each subquery (*i.e.*, 2 * 5 = 10 documents including C^2 query) in SFT-QPlanner and DPO-QPlanner. Additionally, we follow the *associative selection* process, suggested in Lee et al. (2023), to extract relevant evidence paragraphs from retrieved documents. Specifically, we construct FLAN-T5-Large (Chung et al., 2024) trained with Wikipedia-based datasets such as MS-MARCO (Bajaj

[15] `https://serpapi.com/duckduckgo-search-api`

et al., 2018), ELI5 (Fan et al., 2019), ASQA (Stelmakh et al., 2022), and Qasper (Dasigi et al., 2021). The trained language model matches passages in each document with given subqueries and returns an answerability score deciding whether the paired subquery and passage are relevant. We select the top-1 passage for each document as evidence for generating RAG response.



You are invited to participate in a research study that aims to understand and develop user-preferred answering options for common-sense questions.

TIME: Your participation will take approximately 40 minutes.
PAYMENT: You will receive your compensation for study completion.
TASK: You will judge how satisfactory a large language model's answer is, and choose a better answer out of two answers.
RISKS AND BENEFITS: There are no foreseeable risks or benefits to you associated with this study.
PARTICIPANT'S RIGHTS: If you have read this form and have decided to participate in this study, please understand your participation is voluntary and you have the right to withdraw your consent or discontinue participation at any time. The alternative is not to participate. You have the right to refuse to answer particular questions. The results of this research study may be presented at scientific or professional meetings or published in scientific journals. Your individual privacy will be maintained in all published and written data resulting from the study.
CONTACT INFORMATION: If you have any questions, concerns or complaints about this research, its procedures, risks and benefits, contact the Protocol Director, [redacted]

Figure 5: Initial information provided to participants in our human study.