Retrieve-Plan-Generation: An Iterative Planning and Answering Framework for Knowledge-Intensive LLM Generation

Anonymous ACL submission

Abstract

Despite the significant progress of large language models (LLMs) in various tasks, they often produce factual errors due to their limited internal knowledge. Retrieval-Augmented Generation (RAG), which enhances LLMs with external knowledge sources, offers a promising solution. However, these methods can be misled by irrelevant paragraphs in retrieved documents. Due to the inherent uncertainty in LLM generation, inputting the entire document may introduce off-topic information, causing the model to deviate from the central topic and affecting the relevance of the generated content. To address these issues, we propose the Retrieve-Plan-Generation (RPG) framework. RPG generates plan tokens to guide subsequent generation in the plan stage. In the answer stage, the model selects relevant fine-grained paragraphs based on the plan and uses them for further answer generation. This plan-answer process is repeated iteratively until completion, enhancing generation relevance by focusing on specific topics. To implement this framework efficiently, we utilize a simple but effective multi-task prompt-tuning method, enabling the existing LLMs to handle both planning and answering. We comprehensively compare RPG with baselines across 5 knowledge-intensive generation tasks, demonstrating the effectiveness of our approach.

1 Introduction

With the persistent scaling up of training parameters and datasets (Kaplan et al., 2020), large language models (LLMs) (Touvron et al., 2023; Jiang et al., 2023a; Bai et al., 2023; Achiam et al., 2023) have made remarkable advancements, becoming the cornerstone of many Natural Language Processing (NLP) tasks in recent years. Despite improvements in model architecture and the expansion of training data, LLMs still struggle with factual errors (Lyu et al., 2022; He et al., 2022). To address this issue, the Retrieval-Augmented Generation (RAG) system has been introduced (Lewis et al., 2020; Guu et al., 2020). By retrieving external information and incorporating it into the input, the RAG system demonstrates excellent performance in knowledge-intensive tasks.

Question: How do jellyfish function without brains or nervous systems?

Retrieved Documents: Title: Jellyfish, Simple yet Mysterious Ocean Creatures. Jellyfish lack a brain but possess a rudimentary nervous system and neural cells capable of sensing external stimuli such as light and touch[...] They likely coordinate these functions through a nerve ring [...] Jellyfish are invertebrates [...] Some species have remarkably brief lifespans, lasting merely hours. Yet their vitality is remarkably resilient[...] Most strikingly, jellyfish possess an astounding capacity for regeneration [...]

Normal RAG: Despite their seemingly lifeless appearance, jellyfish are among the most wondrous creatures on Earth! Though lacking a brain, jellyfish possess a rudimentary nervous system and neural[...] Their bodies also exhibit a peculiar self-healing ability, rapidly regenerating even when cut into pieces and reassembled. Furthermore, jellyfish can bioluminescent, illuminating the dark [...]

Plan Before Generate: [Plan: Body composition and structure of jellyfish] <p>Jellyfish…Their bodies are almost entirely made up of gelatinous hydrogel tissue, with a water content of more than 98%...</p>Jellyfish are indeed very simple creatures, almost more than 98% of which are made up of water. [Plan: Nervous system of jellyfish] <p>Jellyfish barely have nervous systems…They have primitive nerve cells that help them</p>However, although jellyfish do not have brains, they do have a rough nervous system that can sense external stimuli such as light and touch[...] [Plan: Nerve ring of jellyfish] <p>..</p>Jellyfish coordinate through nerve rings [...]

Figure 1: The retrieval documents contain off-topic paragraphs (highlighted in yellow), causing potential deviations in RAG outputs. By planning first (highlighted in green), selecting relevant fine-grained paragraphs, and then answering, the plan-answer iteration ensures a more consistent and relevant generation.

The most common approach in RAG involves using the user input as a query for a single-time retrieval (Lewis et al., 2020), with LLMs then generating answers based on the retrieved information. However, documents retrieved for input into the LLM are often lengthy, and not all paragraphs may be practically helpful for answering the question. Recent research (Lan and Jiang, 2021; Sun et al., 2023) indicates that off-topic paragraphs can be detrimental to the generation. As the Figure 1 illustrates, due to the inherent uncertainty in the generation process of LLMs, inputting the entire retrieved document can lead to those off-topic paragraphs misleading the model, causing a shift in focus and resulting in content that gradually deviates from the main topic.

Currently, many researchers have acknowledged this issue and have adopted various solutions. Some works (Jiang et al., 2023b; Asai et al., 2023) determine whether retrieval is necessary before generating an answer and input the retrieved document only when required. Self-RAG (Asai et al., 2023) further introduces reflection tokens to evaluate the quality of retrieved documents, thereby excluding irrelevant documents. Despite significant advancements with these methods, their effectiveness diminishes when dealing with longer retrieved texts, particularly those that are generally relevant but contain some irrelevant details. Additionally, when the retrieved documents are too lengthy, it becomes challenging for users to verify the correctness of specific details in the generated content.

We propose that the susceptibility of LLMs to irrelevant content stems from a lack of explicit pre-planning in generating subsequent content. As illustrated in Figure 1, if the model continuously plans the next topic at each step and only focuses on highly relevant paragraphs, it can avoid being misled by irrelevant material during lengthy generation processes. To implement this plan-answer process, we introduce the Retrieve-Plan-Generation (RPG) framework. RPG iterates through two stages: the ***plan stage*** and the ***answer stage***. In the plan stage, the model generates tokens representing upcoming text topics. During the answer stage, the model selects highly on-topic paragraphs from retrieved documents based on these topics, and uses them to generate targeted answers. This iterative process between planning and answering continues until the generation is complete. Unlike traditional full-text input methods, RPG provides detailed control over content generation by focusing on specific topics at each step, ensuring the generation is highly relevant and accurate. Additionally, this fine-grained approach makes it easier for users to verify the correctness of answer details, even when dealing with long documents.

Existing LLMs struggle to effectively integrate both planning and answering capabilities. Since the plan must be incrementally developed during the generation process, relying solely on pre-designed prompts for plan generation is challenging. Additionally, prompts need to guide the model in generating both the plan and the answer based on generated context and relevant paragraphs, which imposes high demands on the model's ability to comprehend complex prompts. Therefore, we prompt ChatGPT to create supervision for plan generation and fine-grained paragraph utilization based on existing datasets (Asai et al., 2023; Yang et al., 2018), then train our model end-to-end on this dataset.

Fully fine-tuning an LLM is resource-intensive and often unnecessary. To balance the learning capabilities of the LLMs with training efficiency, prompt tuning has emerged as a promising method. Given that the input and output formats for planning and answering tasks differ, we adopt a multi-task prompt tuning approach, training two learnable prompt tokens specifically for plan and answer generation. These two prompt tokens share the same soft prompt. During the training stage, each data is simultaneously used for both planning and answering tasks. To train task-specific prompts, we first transform the soft prompt to the corresponding task mode, and then exclude the impact of other parts during loss computation.

Empirical results on 5 tasks, including long-form, multi-hop, and short-form generation, demonstrate that RPG significantly outperforms instruction-tuned LLMs with more parameters and widely adopted RAG approaches. Technical contributions of this paper can be summarized as follows:

- We propose a new framework, RPG, which incorporates an explicit planning stage for LLMs, enhancing generation relevance by focusing on specific topics iteratively.
- We also adopt a simple but effective method that enables existing LLMs to easily configure plan-answer capabilities, adapting to the distinct requirements of these two tasks through multi-task learning.
- Experimental results on 5 tasks demonstrate the superiority of our proposed method over state-of-the-art methods.

2 Related Work

Retrieval-Augmented Generation. Retrieval-Augmented Generation (RAG) (Lewis et al., 2020; Guu et al., 2020) enhances LLMs by retrieving relevant passages, thereby improving both the quality and accuracy of generated content, particularly

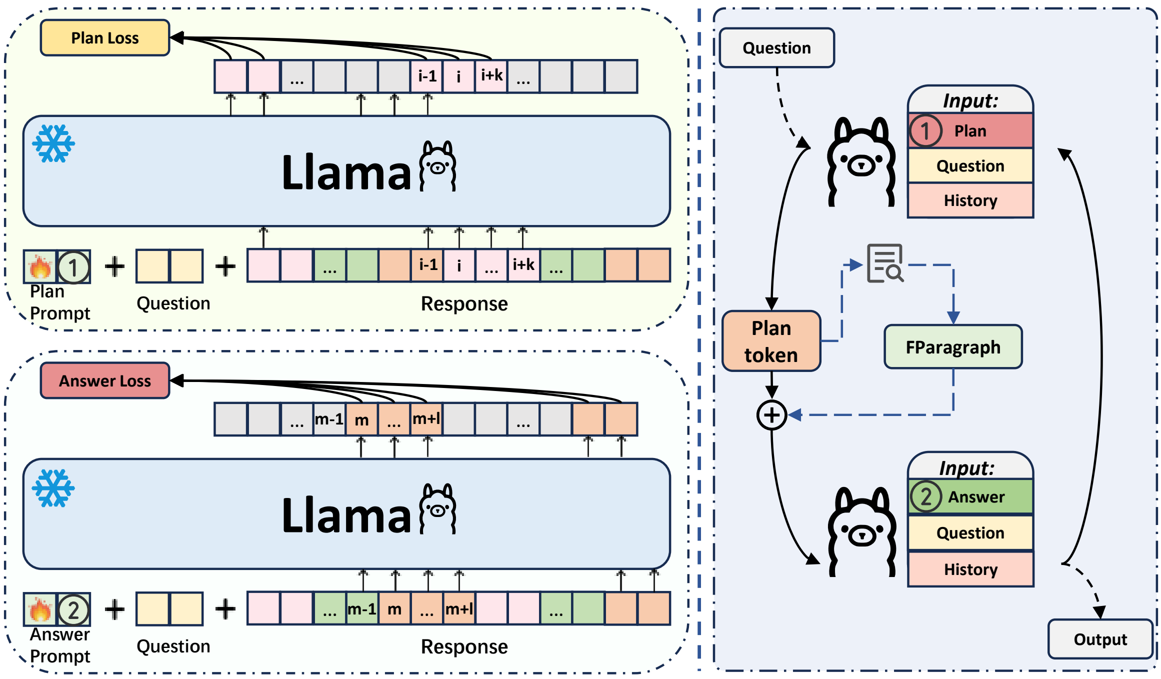

Figure 2: Illustration of the proposed RPG. The left shows the training process, where plan and answer tasks use the same example data, different loss functions, and train two task-specific prompts simultaneously. The right shows the inference process, where the plan-answer process is repeated iteratively until completion.

in knowledge-intensive tasks (Shen et al., 2023; Chen et al., 2023). Early works (Es et al., 2023; Lyu et al., 2024) chose to retrieve once, incorporating a fixed number of retrieved passages with a query into LLMs to generate a response. Recent research indicates that adaptive retrieval, tailored to the demands of LLMs, can further enhance generation. FLARE (Jiang et al., 2023b) uses the generated sentence with a low confidence score as the query to retrieve external knowledge adaptively and then regenerates the current sentence, while Self-RAG (Asai et al., 2023) introduces special tokens allowing the model to adaptively retrieve and reflect the quality of generated content. SuRe (Kim et al., 2024) generates conditional summarizations of retrieval and evaluating them with carefully designed prompts. However, existing approaches may not take full advantage of the planning capabilities of LLMs. Additionally, these methods may struggle to extract relevant content from retrieved passages and are easily influenced by irrelevant information.

Parameter-Efficient Fine-Tuning. Despite the powerful generative capabilities of LLMs, fine-tuning them requires substantial computational resources (Lester et al., 2021; Ding et al., 2022; Liu et al., 2023). To achieve more efficient fine-tuning, parameter-efficient tuning methods have emerged. These methods either fine-tune a small portion of the model parameters or introduce additional learnable parameters without fine-tuning the model itself (Hu et al., 2021; Liu et al., 2021; Ding et al., 2022; Wang et al., 2023). LoRA (Low-Rank Adaptation) (Hu et al., 2021) reduces the number of parameters to be updated by decomposing the weight matrices into low-rank components. Prompt tuning (Liu et al., 2021, 2023) introduces task-specific prompts by concatenating learnable tokens before the input sequence, requiring minimal parameter updates. Multi-task Prompt Tuning (MPT) (Wang et al., 2023) further highlights the commonalities between multi-task learning, suggesting that using a shared soft prompt and task-specific low-rank matrices can yield better results.

3 Methodology

In this section, we first introduce the task definition and basic notation. Then, we provide a comprehensive explanation of the RPG framework from the perspectives of fine-grained dataset construction, training, and inference.

3.1 Task Definition & Notation

Given a user input x, a retriever $\mathcal{R}$ and document corpus $\mathcal{D} = \{d_1, d_2, \dots, d_n\}$, RAG aims to enhance the quality of a language model's (LM's) output y by retrieving relevant passages from $\mathcal{D}$ and incorporating them into the answer. For a query q, the retriever $\mathcal{R}$ can retrieve a list of documents $\mathcal{D}_q = \mathcal{R}(q, \mathcal{D})$ from corpus $\mathcal{D}$.

Vanilla Retrieval Augmented Generation. The most common approach is to use the user input x directly as the query for retrieval, and then generate the complete answer in a single step $y = LM([\mathcal{D}_x, x])$.

Dynamic Retrieval Generation. To aid long-form generation with retrieval, dynamic retrieval generation further refines the RAG approach by dynamically retrieving information according to the model's needs during the generation process. Although dynamic retrieval can reduce the factual errors of LM, the lack of explicit planning may lead to interference from irrelevant information, resulting in the focus shift phenomenon. Based on this fundamental structure, this paper innovatively proposes a two-stage method using the distinct plan and answer stage to achieve generated content with reduced focus shift.

3.2 Method Overview

To enhance the factuality of LLMs and improve topic consistency in long-form generation, LLMs should be capable of generating a preliminary plan to select fine-grained evidence, guiding subsequent content generation on specific topics. Based on this consideration, our RPG framework is designed into two stages: ***plan*** and ***answer***. During the plan stage, the LLM should generate a topic for the upcoming answer, reflecting pre-planned thoughts and guiding the subsequent generation. This approach effectively prevents the output from deviating from the specific topic. In the answer stage, by removing irrelevant information at the sentence level, a foundational denoising capability is achieved. This decouples the processes of filtering and utilizing relevant information during the generation, thereby enhancing the model's ability to leverage fine-grained relevant evidence. Through the iterative alternation of these two stages, the focus shift phenomenon during long text generation can be effectively avoided.

Specifically, to train an LLM end-to-end with both planning and fine-grained evidence utilization capabilities efficiently, multi-task prompt tuning is employed to learn these two tasks synchronously on a dataset we reconstructed. During the inference stage, the LLM iteratively repeats the plan-answer process until the final response is generated. Figure 2 illustrates both the training and inference of the RPG framework.

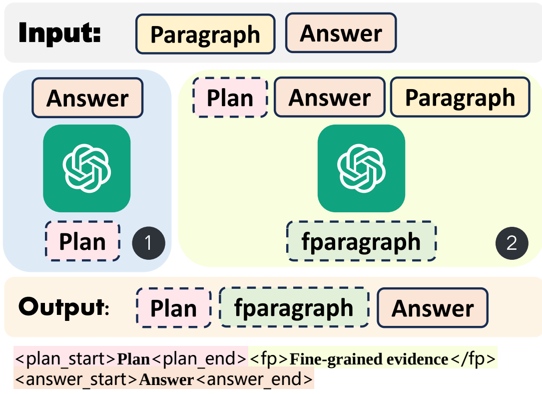

Figure 3: Illustration of the data processing for one of the segments in a sample.

3.3 Dataset Construction

To train the aforementioned LLM, we reconstruct a fine-grained dataset based on the existing Self-RAG (Asai et al., 2023) and HotpotQA (Yang et al., 2018) datasets, where the annotated data has been split into segments with retrieved documents.

Data collection for plan. Since answer segments are specific implementations of an individual's planning at each step, we can treat the intent of these segments as human planning, thereby avoiding topic deviation. As shown in Figure 3, to obtain the intent of each segment, we prompt ChatGPT to summarize the segment and use these summaries as labels for the plan stage. For data that do not require additional retrieved information, we attach <no_info> directly at the beginning of the answer, indicating that no planning is needed and the LLM's inherent ability to answer is sufficient.

Data collection for answer. As mentioned before, the coarse-grained documents provided in existing datasets often contain off-topic paragraphs, which has been shown to be adverse to generation (Yoran et al., 2023). After filtering the paragraphs at the sentence level, we retain only the information related to the plan tokens and the corresponding answer segment for the answer stage training. Specifically, we provide ChatGPT with pre-generated plan tokens, along with corresponding coarse-grained documents and the answer segment. We then require ChatGPT to select sentences related to the plan and answer from the document as fine-grained evidence, which is further used to train the LLM's ability of fine-grained evidence utilization. The answer segments are the labels for the answer stage.

Finally, we collect 50k supervised training data to form a new dataset for RPG training. More details about our dataset are shown in Appendix B. Prompts and examples are shown in Appendix C.

3.4 RPG Training

To efficiently leverage the information within the data, we introduce a multi-task training method for the RPG framework. During the training phase, we utilize different components of the samples, plan and answer, from the constructed dataset to train the model. Simultaneously, we train two task-specific learnable prompts with different loss functions. This approach enables the frozen LLM to acquire planning and answering capabilities without requiring any modifications to the model itself.

As shown in Figure 2, to achieve more parameter-efficient fine-tuning, we opt to freeze the LLM and train the additional continuous prompt vectors prepended to the input. Recent research (Wang et al., 2023) indicates that commonalities exist across various tasks, paving the way for more efficient prompt tuning. Following them, we first employ a soft prompt P^* as the shared prompt across plan and answer tasks. To adapt to the distinct requirements of these two tasks, we further utilize two different low-rank matrices, W_{plan} and W_{ans}, to transform the soft prompt to the specific task mode. The task prompts for plan and answer generation task are parameterized as follows:

$$P_{task} = P^* \circ W_{task} = P^* \circ (u_{task} \otimes v_{task}^T), \tag{1}$$

where $\circ$ denotes the Hadamard product between two matrices, and $task \in \{plan, ans\}$ denotes the specific generation task.

To enhance the efficiency of multi-task training, we utilize different components of the samples to simultaneously train the plan prompt and the answer prompt. Specifically, we adopt to mask different parts of the same data instance to guide the learning of corresponding tasks. For the ***plan stage*** training, tokens other than the plan tokens in the ground truth are masked, guiding the LLM to focus solely on plan generation. Similarly, for the ***answer stage***, tokens that are not part of the answer are excluded from the loss calculation. For formal expression, the conditional language modeling objective $\mathcal{L}_{plan}$ and $\mathcal{L}_{ans}$ are employed to optimize our model $\mathcal{M}$ in two stages:

$$\mathcal{L}_{plan} = -\sum_{y_i \in plan} \log P(y_i|x_i; \Theta, P_{plan}), \tag{2}$$

$$\mathcal{L}_{ans} = -\sum_{y_i \in ans} \log P(y_i|x_i; \Theta, P_{ans}), \tag{3}$$

where P_{plan} and P_{ans} are learnable. During training, we combine the two loss functions and optimize the model parameters simultaneously.

3.5 RPG Inference

Algorithm 1 RPG Inference

Require: Generator LLM $\mathcal{M}$, Retriever $\mathcal{R}$, Large-scale passage collections $\mathcal{D} = \{d_1, \ldots, d_N\}$, Task Prompts P_{plan}, P_{ans}

Input: user input x and retrieved relevant passages $\mathcal{D}_x = \mathcal{R}(x, \mathcal{D})$, **Output:** response y
Initialize the response $y \leftarrow \emptyset$
$\mathcal{M}$ predicts plan $\mathcal{P}$ given (P_{plan}, x)
if $\mathcal{P}$ == `<no_info>` **then**
    $\mathcal{M}$ generates y given (P_{ans}, x)
else
    while $\mathcal{M}$ has not generated the `<EOS>` token **do**
        Select relevant paragraphs e given $(\mathcal{D}_x, \mathcal{P})$
        $\mathcal{M}$ predicts y_t given $(P_{ans}, x, e, y_{<t})$
        $\mathcal{M}$ predicts plan $\mathcal{P}$ given $(P_{plan}, x, y_{<t})$
        Append y_t to y
    end while
end if
return y

Figure 2 and Algorithm 1 presents an overview of RPG at inference. During the inference phase, the RPG framework enhances response quality by iteratively invoking the plan-answer capability. This approach not only provides additional knowledge to the LLM but also ensures topic consistency. To reduce costs, the bge-reranker (Xiao et al., 2023) is employed instead of ChatGPT to select fine-grained on-topic paragraphs during the inference phase. Specifically, for every user input x and retrieved passages $\mathcal{D}_x$, the LLM $\mathcal{M}$ first determines whether additional information is needed. If $\mathcal{M}$ generates `<no_info>`, the LLM $\mathcal{M}$ predicts the output y directly using prompt P_{ans} and input x. In other cases, relevant information about plan tokens $\mathcal{P}$ in retrieved passages is selected as fine-grained paragraphs to supplement $\mathcal{M}$ with external

LLMs	ASQA (rg)	ASQA (mau)	ELI5 (rg)	ELI5 (mau)
SOTA LLMs				
ChatGPT	36.2	68.8	**22.8**	32.6
Ret-ChatGPT	**39.9**	79.7	20.6	**57.2**
Baselines without retrieval				
Llama2 $_{7B}$	15.3	19.0	18.3	32.4
Alpaca $_{7B}$	29.4	61.7	-	-
Llama2 $_{13B}$	12.4	16.0	18.2	41.4
Alpaca $_{13B}$	32.0	70.6	-	-
Baselines with retrieval				
Llama2 $_{7B}$	22.1	32.0	18.6	35.3
Alpaca $_{7B}$	33.3	57.9	-	-
Llama2-FT $_{7B}$	35.8	51.2	-	-
Llama2 $_{13B}$	20.5	24.7	18.6	42.3
Alpaca $_{13B}$	36.7	56.6	-	-
Self-RAG $_{7B}$	35.7	74.3	17.9	35.6
Self-RAG $_{13B}$	37.0	71.6	-	-
RPG $_{7B}$	**37.6**	**84.4**	**19.1**	**46.4**

Table 1: The experimental results on Long-form generation tasks. Bold numbers indicate the best performance except ChatGPT.

knowledge. Furthermore, the LLM $\mathcal{M}$, using answer prompt P_{ans}, then incorporates fine-grained paragraphs into the generation of the next output segment y_t. This segment y_t is subsequently appended to y. The plan-answer process is repeated until the <EOS> token is generated, at which point y is output as the final answer.

4 Experiments

4.1 Experiment Setup

To validate the effectiveness of our Plan-Retrieve-Generation framework, we conduct in-depth experiments on 5 carefully selected knowledge-intensive tasks. Aligning with the previous work (Asai et al., 2023), we conduct zero-shot evaluations and utilize metrics focused on assessing the correctness, factuality, and fluency of outputs.

4.1.1 Tasks and Datasets

Long-form generation tasks. The long-form QA tasks aim to generate comprehensive answers to questions seeking complex information, which is a primary application scenario for our model. And evaluations of these tasks can serve as evidence to the frameworks' capability of generating on-topic and comprehensive answers. We utilize ASQA (Stelmakh et al., 2022) and ELI5 (Fan et al., 2019) as our testbed, where inputs are ambiguous questions with multiple interpretations, and outputs are expected to address them comprehensively. Following Self-RAG (Asai et al., 2023) and ALCE (Gao et al., 2023), we use ROUGE (Lin, 2004) and MAUVE (Pillutla et al., 2021) for correctness and fluency evaluations.

Multi-hop generation tasks. A multi-hop QA task aims to test reasoning and inference skills by requiring a model to read multiple paragraphs and answer a given question. We use the 2WikiMultiHopQA (Ho et al., 2020) dataset and adopt the F1 score as the metric.

Short-form generation tasks. The short-form QA tasks aim to generate precise answers for users, which evaluate the model's ability to effectively leverage retrieved information to response precisely. We use two open-domain QA datasets, PopQA (Mallen et al., 2022) and PubHealth (Zhang et al., 2023), where models need to answer arbitrary questions about factual knowledge. We process these two datasets following (Asai et al., 2023).

4.1.2 Baselines

Our training dataset is derived from Self-RAG and HotpotQA, where each sample is divided into planning and answering segments using ChatGPT. To ensure a fair comparison, we select baseline models that are fundamentally consistent with Self-RAG and categorize them into three major groups.

Baselines without retrieval. To explore the specific impact of external knowledge on model performance, several retrieval-free baselines are established. We evaluate the open-source models Llama2$_{7B,\ 13B}$ and Alpaca$_{7B,\ 13B}$ (Touvron et al., 2023), which have shown outstanding performance on various tasks.

Baselines with retrieval. We further set up baseline models with retrieval, covering the standard RAG systems. The standard RAG generates content by merging the query and retrieved documents into the input. We also compare the full-parameter fine-tuned version of Llama2-FT based on Self-RAG train data. And we evaluate the Self-RAG model, which enhances the standard RAG by introducing dynamic retrieval and reflection tokens.

ChatGPT-Based baselines. Lastly, we conduct a comparison with the SOTA in the field of LLMs:

LLMs	Multi-Hop 2Wiki (F1)	Short-form PopQA (acc)	Pub (acc)
SOTA LLMs			
ChatGPT	24.8	29.3	70.1
Ret-ChatGPT	32.8	50.8	54.7
SuRe $_{\text{GPT}}$	38.1	-	-
Baselines without retrieval			
Llama2 $_{\text{7B}}$	18.9	14.7	34.2
Alpaca $_{\text{7B}}$	-	23.6	49.8
Llama2 $_{\text{13B}}$	20.2	14.7	29.4
Alpaca $_{\text{13B}}$	-	24.4	55.5
Baselines with retrieval			
Llama2 $_{\text{7B}}$	21.0	38.2	30.0
Alpaca $_{\text{7B}}$	-	46.7	40.2
Llama2-FT $_{\text{7B}}$	-	48.7	64.3
Llama2 $_{\text{13B}}$	31.2	45.7	30.2
Alpaca $_{\text{13B}}$	-	46.1	51.1
SuRe $_{\text{Llama2 7B}}$	20.6	-	-
Self-RAG $_{\text{7B}}$	25.1	54.9	72.4
Self-RAG $_{\text{13B}}$	-	55.8	**74.5**
RPG $_{\text{7B}}$	**33.6**	**56.0**	73.4

Table 2: The experimental results on Multi-Hop and Short-form generation tasks. Bold numbers indicate the best performance except ChatGPT.

ChatGPT and Ret-ChatGPT (ChatGPT with retrieval passage). As a leading LLM, ChatGPT has demonstrated exceptional performance across multiple domains, providing a strong comparative benchmark for our model.

4.1.3 Implementation Details

Training. As mentioned before, our training data is reconstructed from the Self-RAG dataset. We adopt Llama2$_{\text{7B}}$ as our foundational LLM, and use the prompt tuning implementation of the Huggingface PEFT (Mangrulkar et al., 2022) library to fine-tune LLama2$_{\text{7B}}$ on 4 Nvidia A6000 GPUs.

Inference. During inference, the plan and answer stages alternate, using a simple greedy decoding strategy. The plan phase has a token limit of 30, and the answering phase is 100. For short-form QA, the model only completes one plan-answer cycle. For long-form and multi-hop QA tasks, the model alternates between planning and answering until it generates a termination symbol or reaches the operation limit (3 in this paper). In multi-hop QA, a special "[Combine]" symbol indicates that the model will summarize the previous content to produce a concise answer. For the retriever model, we use Contriever-MS MARCO for PopQA, PubHeath, and ASQA datasets, and BM25 for the 2WikiMultiHop datasets, aligning with all baselines.

4.2 Experiment Results

Long-form generation. Our model demonstrates brilliant performance in the domain of long-form generation, which is the primary application scenario for our model. As Table 1 displayed, the experimental results demonstrate that our model has achieved a significant improvement in long-form generation performance with only a slight tuning of 0.3 billion parameters. Notably, our model outperforms the prior SOTA Self-RAG. Specifically, on the ASQA dataset, our model outperforms Self-RAG by 2 points on the ROUGE metric, which measures the correctness and comprehensiveness of long-form generation. Additionally, on MAUVE, a newly introduced metric for evaluating the fluency and coherence of model-generated text, our model significantly outperforms the Self-RAG model by more than 10 points. Even when compared to the current SOTA model, ChatGPT with retrieved knowledge, our model achieves comparable results. Similar findings are also observed in the ELI5 dataset.

These results underscore our model's strong capabilities in long-form generation tasks, demonstrating the comprehensiveness and relevance of our model's responses. The iterative alternation between the planning and answering phases ensures that the generated text remains on-topic and coherent. Our approach not only enhances fluency but also maintains factual accuracy, further highlighting the superiority of our method.

Multi-hop generation. For multi-hop generation tasks, the model needs to integrate all generated information to provide a concise answer. Experimental results in Table 2 indicate that our RPG framework significantly outperforms other Llama-based baseline models, demonstrating the benefit of pre-planning and utilizing fine-grained evidence for reasoning. While the GPT-based SuRe (Kim et al., 2024) model performs better than ours, the Llama-based SuRe model performs poorly due to its dependence on rewriting retrieved content, a process reliant on LLM's capabilities. In contrast, our model avoids this rewriting process and still achieves exceptional performance on multi-hop datasets.

Variations	Pub (acc)	2Wiki (F1)	ASQA (rg)
RPG	**73.4**	**33.6**	**37.6**
Training			
No Plan	69.1	27.4	32.0
No Multi-task Learning	70.1	23.1	34.1
Inference			
No Retrieval	72.3	27.0	32.4
No Paragraph Selection	72.3	30.2	34.8

Table 3: Ablations in training and inference.

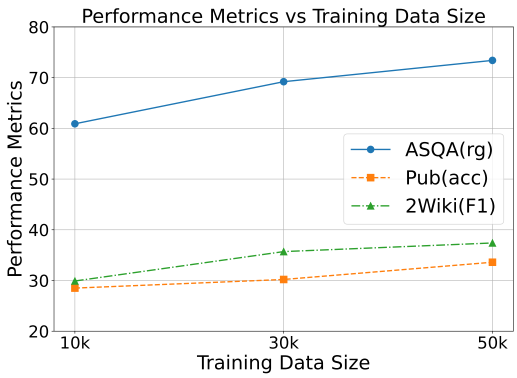

Figure 4: Training scale analysis.

Short-form generation. Although short-form generation is not the primary application scenario of our model, we still demonstrate its performance in this context to prove its versatility and applicability. In some short-form generation tasks, especially on the Pub dataset, we find that retrieved content is not always effective. In fact, retrieval-augmented ChatGPT often underperforms compared to its non-retrieval-augmented counterpart due to the incorporation of irrelevant information. By focusing on the relevance of retrieved content and excluding irrelevant details, our model shows progress in various short-form generation tasks.

4.3 Ablations

As shown in Table 3, we conduct a comprehensive ablation study on the RPG framework to clarify which factors play a decisive role in the training and inference processes.

Training Phase. We investigate the impact of removing the planning phase on model performance. By eliminating all plan texts from the training dataset and using prompt tuning to train the model with only answer texts, we observe a significant drop in performance across all three tasks. In long-form generation, the absence of planning caused the model to deviate from the topic. In short-form generation, unscreened retrieved texts were not always beneficial. Thus, the planning phase is crucial for maintaining the relevance of generated content.

Furthermore, we investigate the differences between fine-tuning a model with uniform learnable prompt tokens for both plans and answers versus using distinct tokens for each. Results show that uniform tokens diminished performance in both long-term and short-term generation tasks, suggesting that planning and answering function as separate tasks. Thus, it is more appropriate to use multi-task learning to train LLMs for both planning and answering capabilities.

Additionally, we study the model's performance with varying scales of training datasets as Figure 4 displayed. The results show that performance gradually improves as the dataset size increases. We believe further expanding the training data will continue to enhance the model's performance.

Inference Phase. In the inference phase, we assess the impact of retrieval on model performance. Results show that retrieval is crucial for long-form generation tasks, which require comprehensive answers. Without retrieval, generating complete answers is significantly more challenging. Conversely, for short-term generation tasks, retrieval has a minor impact, since these tasks may typically do not require extensive knowledge.

Additionally, we examine the effects on model performance when using retrieved passages directly. The results show a significant decline in performance across all tasks, highlighting the detrimental impact of off-topic paragraphs on the quality of generated outputs.

5 Conclusion

In this paper, we propose a Retrieve-Plan-Generation (RPG) framework, which integrates an explicit plan stage into the lengthy generation process. By generating plan tokens, the model is guided to selectively utilize retrieved paragraphs. The iterative alternation between plan and answer stages ensures that the generated content remains relevant to the topic. To implement this framework, we adopt an efficient multi-task fine-tuning method that equips existing models with both planning and answering capabilities. Experimental results demonstrate that RPG outperforms state-of-the-art models across five tasks, validating the effectiveness of our approach.

6 Limitations

Due to computational resource constraints, we only present the specific implementation of the RPG framework under the Llama2$_{7B}$, without exploring further experiments on larger models, such as Llama2$_{13B}$, Llama2$_{70B}$. Additionally, due to the API costs associated with accessing ChatGPT, we conducted experiments solely on a 50k reconstructed dataset, without collecting and analyzing more extensive data to provide more experimental results on larger datasets.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2023. Self-rag: Learning to retrieve, generate, and critique through self-reflection. *arXiv preprint arXiv:2310.11511*.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Yuyan Chen, Qiang Fu, Yichen Yuan, Zhihao Wen, Ge Fan, Dayiheng Liu, Dongmei Zhang, Zhixu Li, and Yanghua Xiao. 2023. Hallucination detection: Robustly discerning reliable answers in large language models. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 245–255.

Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, et al. 2022. Delta tuning: A comprehensive study of parameter efficient methods for pre-trained language models. *arXiv preprint arXiv:2203.06904*.

Shahul Es, Jithin James, Luis Espinosa-Anke, and Steven Schockaert. 2023. Ragas: Automated evaluation of retrieval augmented generation. *arXiv preprint arXiv:2309.15217*.

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.

Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. 2023. Enabling large language models to generate text with citations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6465–6488, Singapore. Association for Computational Linguistics.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *International conference on machine learning*, pages 3929–3938. PMLR.

Hangfeng He, Hongming Zhang, and Dan Roth. 2022. Rethinking with retrieval: Faithful large language model inference. *arXiv preprint arXiv:2301.00303*.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. *arXiv preprint arXiv:2011.01060*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023a. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Zhengbao Jiang, Frank F Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023b. Active retrieval augmented generation. *arXiv preprint arXiv:2305.06983*.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Jaehyung Kim, Jaehyun Nam, Sangwoo Mo, Jongjin Park, Sang-Woo Lee, Minjoon Seo, Jung-Woo Ha, and Jinwoo Shin. 2024. Sure: Summarizing retrievals using answer candidates for open-domain qa of llms. *arXiv preprint arXiv:2404.13081*.

Yunshi Lan and Jing Jiang. 2021. Modeling transitions of focal entities for conversational knowledge base question answering. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3288–3297.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Lam Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2021. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. *arXiv preprint arXiv:2110.07602*.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2023. Gpt understands, too. *AI Open*.

Yuanjie Lyu, Zhiyu Li, Simin Niu, Feiyu Xiong, Bo Tang, Wenjin Wang, Hao Wu, Huanyong Liu, Tong Xu, and Enhong Chen. 2024. Crud-rag: A comprehensive chinese benchmark for retrieval-augmented generation of large language models. *arXiv preprint arXiv:2401.17043*.

Yuanjie Lyu, Chen Zhu, Tong Xu, Zikai Yin, and Enhong Chen. 2022. Faithful abstractive summarization via fact-aware consistency-constrained transformer. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 1410–1419.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2022. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. *arXiv preprint arXiv:2212.10511*.

Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, and Benjamin Bossan. 2022. Peft: State-of-the-art parameter-efficient fine-tuning methods. `https://github.com/huggingface/peft`.

Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernández Ábrego, Ji Ma, Vincent Y Zhao, Yi Luan, Keith B Hall, Ming-Wei Chang, et al. 2021. Large dual encoders are generalizable retrievers. *arXiv preprint arXiv:2112.07899*.

Krishna Pillutla, Swabha Swayamdipta, Rowan Zellers, John Thickstun, Sean Welleck, Yejin Choi, and Zaid Harchaoui. 2021. Mauve: Measuring the gap between neural text and human text using divergence frontiers. *Advances in Neural Information Processing Systems*, 34:4816–4828.

Xinyue Shen, Zeyuan Chen, Michael Backes, and Yang Zhang. 2023. In chatgpt we trust? measuring and characterizing the reliability of chatgpt. *arXiv preprint arXiv:2304.08979*.

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.

Hao Sun, Hengyi Cai, Bo Wang, Yingyan Hou, Xiaochi Wei, Shuaiqiang Wang, Yan Zhang, and Dawei Yin. 2023. Towards verifiable text generation with evolving memory and self-reflection. *arXiv preprint arXiv:2312.09075*.

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

Zhen Wang, Rameswar Panda, Leonid Karlinsky, Rogerio Feris, Huan Sun, and Yoon Kim. 2023. Multitask prompt tuning enables parameter-efficient transfer learning. *arXiv preprint arXiv:2303.02861*.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. C-pack: Packaged resources to advance general chinese embedding.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Ori Yoran, Tomer Wolfson, Ori Ram, and Jonathan Berant. 2023. Making retrieval-augmented language models robust to irrelevant context. *arXiv preprint arXiv:2310.01558*.

Tianhua Zhang, Hongyin Luo, Yung-Sung Chuang, Wei Fang, Luc Gaitskell, Thomas Hartvigsen, Xixin Wu, Danny Fox, Helen Meng, and James Glass. 2023. Interpretable unified language checking. *arXiv preprint arXiv:2304.03728*.

A More PRG Implementation Details

Training As previously mentioned, our training data is structured based on the Self-RAG dataset. During the training phase, we utilize the Llama2$_{7B}$ as our foundational language model. For the retriever model, we have selected the readily accessible Contriever-MS MARCO for the PopQA, Pub, and ASQA datasets, and the BM25 algorithm for the 2Wiki datasets, aligning with the baselines

Inference During the inference process, the planning and answering stages alternate, and we have employed a simple greedy decoding strategy for both. In the planning phase, we set a maximum token generation limit of 30, while in the answer phase, it is 100. As for the retrieved documents, by default, we use the top five documents ranked by Contriever-MS MARCO (Izacard et al., 2021); For ASQA, we utilized the top five documents selected by the authors from GTR-XXL (Ni et al., 2021), which is done to ensure a fair comparison among all baseline models. In short-form QA, the model executes a single planning and answering cycle. Conversely, in long-form QA tasks, the model alternates between planning and answering multiple times until it generates a termination symbol or reaches the limit of operations(3 in this paper). Multi-hop QA follows a similar approach to long-form QA. However, there is a minor difference: as the generation process nears completion, our model generates a special "[Combine]" symbol. This indicates that the model will then summarize the previously generated content and ultimately produce a concise answer to the original question.

B Statistical information of the Dataset

In this section, we provide a detailed discussion of the statistical information and relevant details of the dataset. The statistical information of the experimental data is presented in Table 4, with additional statistics on the dataset's Plan information shown in Table 5.

C Prompts for Dataset construction and Examples

In this section, we provide a detailed explanation of the construction methods for each dataset. We first introduce the instructions used for dataset construction and then provide corresponding examples for each dataset.

Instructions for Plan Generation of Short-form QA

Plan Generation:
Instructions:
Extract the body of the statement from the question into a Plan token. The plan token should be like [Plan: XX].
Input: which company Javed Afridi is best known as CEO?
Output: [Plan: Javed Afridi best known company].
Input: *a question*
Output:

Figure 5: Instructions for Constructing Plan Generation Datasets of Short-form QA

To construct our own dataset, we utilize `gpt-3.5-turbo-0125` to generate comprehensive annotations leveraging existing datasets and few-shot examples. Given the straightforward nature of the short-form questions, we prompt ChatGPT to summarize their statements as the Plan, which is outlined in Figure 5. We apply this method to Natural Questions, FEVER, OpenBoookQA, and Arc-Easy. ASQA consists of numerous ambiguous questions, where each problem within the annotated dataset is further divided into multiple sub-problems post artificial disambiguation. These segments address specific parts of the question, and due to their close resemblance, ChatGPT may generate very similar topics based on answer summaries. ChatGPT should identify which sub-problems the current answer corresponds to, and then summarize these sub-problems into a statement. The process is guided by prompts detailed in Table 6. As ShareGPT does not encounter many ambiguous questions, for each part of the answer, we directly prompt ChatGPT to summarize the current segment's topic based on the provided answer context as the label for the Plan Generation. Detailed information is provided in Table 7. For HotpotQA, since there is sufficient evidence in the annotated data and the question only needs two jumps at most, we believe that ChatGPT is sufficient to give good planning based on the question and answer. The instructions are shown in Table 9. Prompts used for fine-grained evidence selection are shown in Table 8. Examples of our dataset can be found in Table 10, Table 11, and Table 12.

Dataset name	**Category**	**Data source**	**# of instances**
ShareGPT	Instruction-following	Open-Instruct	13,095
Natural Questions	Knowledge-intensive	KILT	15,226
FEVER	Knowledge-intensive	KILT	9,665
OpenBoookQA	Knowledge-intensive	HF Dataset	4,699
Arc-Easy	Knowledge-intensive	HF Dataset	1847
ASQA	Knowledge-intensive	ASQA	3,564
HotpotQA	Knowledge-intensive	HotpotQA	3830

Table 4: Dataset statistics

Dataset name	**Data source**	**Avg. # of Plan**	**% of Plan=`True`**
ShareGPT	Open-Instruct	3.939	70.3
Natural Questions	KILT	0.877	87.7
FEVER	KILT	0.634	63.4
OpenBoookQA	HF Dataset	0.023	2.3
Arc-Easy	HF Dataset	0.108	10.8
ASQA	ASQA	1.916	91.6
HotpotQA	HotpotQA	1.338	77.8

Table 5: Detailed plan statistics of dataset

Instructions for Plan Generation of ASQA

Plan Generation:

Instructions:

Given several short qa-pairs and a sentence, you need to decide which qa-pair is this sentence relevant to. Always cite for any factual claim. When citing several search results, use `[1][2][3]`. If multiple qa-pairs support the sentence, only cite a minimum sufficient subset of the qa-pairs.

QA-Pairs:

`[0]` Where Haier Pakistan is located? Pakistan.

`[1]` When was Haier Pakistan established? 2000.

Sentence:

Established in 2000, it is a subsidiary of the Chinese multinational group Haier.

Out: `[1]`

QA-Pairs:

`[0]` When does episode 42 of bunk'd come out? May 24, 2017.

`[1]` When does episode 41 of bunk'd come out?? April 28, 2017.

`[2]` When does episode 40 of bunk'd come out? April 21, 2017.

Sentence:

The new bunk'd episode 41 comes out on April 21, 2017, episode 42 comes out on April 28, 2017 and episode 42 is due to come out on May 24, 2017.

Out: `[0][1][2]`

QA-Pairs:

your qa-pairs

Sentence:

your sentence

Out:

Given a number of questions, you need to summarize them as concisely and accurately as possible into one question, avoiding missing information about each question. You don't have to answer these questions.

Questions:

`[0]` first question

`[1]` second question

...

Output:

Table 6: Instructions for Plan Generation of ASQA

Instructions for Plan Generation of ShareGPT

Plan Generation:

Instructions:

Generate appropriate Plan token in the following format: [Plan: xx], for each [Plan] based on relevant context. Be sure always generate a Plan Token for each [Plan] in order, Keep the details to be as different as possible from other Plan tokens. Do not generate a Plan Token where there is no [Plan].

Input: AB is famous for his work in Parkistan Haier.[Plan] Established in 2000, it is a subsidiary of the Chinese multinational group Haier.

Output:AB is famous for his work in Parkistan Haier.[Plan: Parkistan Haier establish time] Established in 2000, it is a subsidiary of the Chinese multinational group Haier.

Input: answer segment

Output:

Table 7: Instructions for Plan Generation for each answer segment of ShareGPT

Instructions for Fine-grained evidence selection

Fine-grained evidence selection:

Instructions:

Write an accurate, engaging, and concise answer for the given question answer pair using only the provided search results (some of which might be irrelevant) and cite them properly. Use an unbiased and journalistic tone. Always cite for any factual claim. When citing several search results, use `[1][2][3]`. If multiple documents support the sentence, only cite a minimum sufficient subset of the documents.

Question: When was Haier Pakistan established?

Answer: 2000.

`[0]` Haier Pakistan is a consumer electronics and home appliances company in Pakistan.

`[1]` Established in 2000, it is a subsidiary of the Chinese multinational group Haier.

`[2]` It is one of the largest companies in Pakistan's home appliances market, in terms of sales and revenues generated.

Out: `[1]`

Ouestion: question

Answer: answer

`[0]` first evidence

`[1]` second evidence

...

Out:

Table 8: Instructions for Fine-grained evidence selection for each answer segment

Instructions for HotpotQA

Plan Generation:

Instructions:

Given a question and corresponding short answer. Expand the short answer to an accurate, fine-grained, and concise answer with thinking steps for the given question using only the provided search results (some of which might be irrelevant) and cite them properly. During the generation, make sure the plan token in answers start with the question and work their way up logically from the answers you already have. Use an unbiased and journalistic tone. Always cite for any factual claim. Cite at most one evidence in each sentence. If multiple documents support the sentence, only cite the first one.

Question: In what year was the company, for which Javed Afridi is best known as CEO, established?

Answer: 2000.

Evidence:

`[0]` Established in 2000, Haier Pakistan is a subsidiary of the Chinese multinational group Haier.

`[1]` Javed Afridi is best known as the CEO of Haier Pakistan and owner of MG Motors Pakistan.

Out: [Plan: Javed Afridi best known company]Javed Afridi is best known as the CEO of Haier Pakistan`[1]`,[Plan: Haier Pakistan establish]which was established in 2000.`[0]`

As mentioned before, the first plan token should be generated from question [Plan: Javed Afridi best known company], considering the answer already generated and further plan the establish time of Haier Pakistan.

Question: Where are Steph Curry and Lebron James both from?

Answer: America.

Evidence:

`[0]` Stephen Curry is a professional American basketball player with the Golden State Warriors..

`[1]` Lebron James is a professional American basketball player with the Lakers.

Out: [Plan: Steph Curry from]Steph Curry is from America`[0]`,[Plan: Lebron James from]Lebron James is also from America.`[1]`

Question: question

Answer: answer

Evidence: your evidences

Out:

Table 9: Instructions for HotpotQA

Input: What harmful phenomenon can cause atmospheric warming? A: gasses with pollution B: acid rain C: nuclear war D: melting ice caps (*OpenBoookQA*)
Output: <plan_start>Harmful phenomenon causing atmospheric warming<plan_end><paragraph>Asian brown cloud past century. The report suggested that air pollution regulations can have large amplifying effects on global warming. Another major impact is on the polar ice caps. Black carbon (soot) in the Asian Brown Cloud may be reflecting sunlight and dimming Earth below but it is warming other places by absorbing incoming radiation and warming the atmosphere and whatever it touches. Black carbon is three times more effective than carbon dioxide—the most common greenhouse gas—at melting polar ice and snow. Black carbon in snow causes about three times the temperature change as carbon dioxide in the atmosphere. On snow—even at concentrations</paragraph><fparagraph>The report suggested that air pollution regulations can have large amplifying effects on global warming.</fparagraph><answer_start>A<answer_end>
Input:how long does a president have secret service protection (*Natural Questions*)
Output: <plan_start>president secret service protection duration<plan_end><paragraph>Former Presidents Act health insurance under the Federal Employees Health Benefits Program; a GSA legal opinion ruled Jimmy Carter ineligible. Former presidents were entitled from 1965 to 1996 to lifetime Secret Service protection, for themselves, spouses, and children under 16. A 1994 statute, (), limited post-presidential protection to ten years for presidents inaugurated after January 1, 1997. Under this statute, Bill Clinton would still be entitled to lifetime protection, and all subsequent presidents would have been entitled to ten years' protection. On January 10, 2013, President Barack Obama signed the Former Presidents Protection Act of 2012, reinstating lifetime Secret Service protection for his</paragraph><fparagraph>Former presidents were entitled from 1965 to 1996 to lifetime Secret Service protection, for themselves, spouses, and children under 16. Under this statute, Bill Clinton would still be entitled to lifetime protection, and all subsequent presidents would have been entitled to ten years' protection. </fparagraph><answer_start>lifetime<answer_end>
Input: Is the following statement correct or not? Say true if it's correct; otherwise say false.## Input: Evan Goldberg is a Canadian director. (*FEVER*)
Output: <plan_start>Evan Goldberg is a Canadian director.<plan_end><paragraph>Evan Goldberg Evan Goldberg Evan Goldberg (born May 11, 1982) is a Canadian screenwriter, film producer, and director. He has collaborated with his childhood friend Seth Rogen in several films, including "Superbad" (2007) (which they first conceived as teenagers), "Pineapple Express" (2008), "This Is the End" (2013) (their directorial debut), and "The Interview" (2014). Goldberg was born in Vancouver, British Columbia, to a Jewish family. He was raised in Marpole. He attended Point Grey Secondary School (where he met Rogen) and McGill University, and is married to Lisa (Yadavaia) Goldberg. Goldberg started his writing career joining the staff of "Da Ali G</paragraph><fparagraph>Evan Goldberg Evan Goldberg Evan Goldberg (born May 11, 1982) is a Canadian screenwriter, film producer, and director.</fparagraph><answer_start>true<answer_end>
Input: Given four answer candidates, A, B, C and D, choose the best answer choice. ## Input: Darwin's theory that animal species can change over time was inspired by his research on which set of islands? A: the Philippine Islands B: the Virgin Islands C: the Hawaiian Islands D: the Galapagos Islands(*Arc-Easy*)
Output: <plan_start>Darwin's theory animal species change over time inspired research islands<plan_end><paragraph>Speciation tendency of small, isolated genetic pools to produce unusual traits. Examples include insular dwarfism and the radical changes among certain famous island chains, for example on Komodo. The Galápagos Islands are particularly famous for their influence on Charles Darwin. During his five weeks there he heard that Galápagos tortoises could be identified by island, and noticed that finches differed from one island to another, but it was only nine months later that he reflected that such facts could show that species were changeable. When he returned to England, his speculation on evolution deepened after experts informed him that these were</paragraph><fparagraph>The Galápagos Islands are particularly famous for their influence on Charles Darwin.</fparagraph><answer_start>D<answer_end>

Table 10: Examples for short-form datasets

Input: Answer the following question. The question may be ambiguous and have multiple correct answers, and in that case, you have to provide a long-form answer including all correct answers.## Input: Where is superoxide dismutase found in the body?(*ASQA*)

Output: <plan_start>different types of superoxide dismutase found in the body<plan_end><paragraph>Reactive oxygen species nearly all cells exposed to oxygen. In mammals and most chordates, three forms of superoxide dismutase are present. SOD1 is located primarily in the cytoplasm, SOD2 in the mitochondria and SOD3 is extracellular. The first is a dimer (consists of two units), while the others are tetramers (four subunits). SOD1 and SOD3 contain copper and zinc ions, while SOD2 has a manganese ion in its reactive centre. The genes are located on chromosomes 21, 6, and 4, respectively (21q22.1, 6q25.3 and 4p15.3-p15.1). The SOD-catalysed dismutation of superoxide may be written with the following half-reactions: where M = Cu ("n" =</paragraph><fparagraph>SOD1 is located primarily in the cytoplasm, SOD2 in the mitochondria and SOD3 is extracellular.The genes are located on chromosomes 21, 6, and 4, respectively (21q22.1, 6q25.3 and 4p15.3-p15.1).</fparagraph><answer_start>Three forms of superoxide dismutase are present in humans, in all other mammals, and in most chordates, SOD1, SOD2, and SOD3, each located in a different part of the body.<answer_end><plan_start>different types of superoxide dismutase (SOD1, SOD2, SOD3) found in the body<plan_end><paragraph>Reactive oxygen species nearly all cells exposed to oxygen. In mammals and most chordates, three forms of superoxide dismutase are present. SOD1 is located primarily in the cytoplasm, SOD2 in the mitochondria and SOD3 is extracellular. The first is a dimer (consists of two units), while the others are tetramers (four subunits). SOD1 and SOD3 contain copper and zinc ions, while SOD2 has a manganese ion in its reactive centre. The genes are located on chromosomes 21, 6, and 4, respectively (21q22.1, 6q25.3 and 4p15.3-p15.1). The SOD-catalysed dismutation of superoxide may be written with the following half-reactions: where M = Cu ("n" =</paragraph><fparagraph>SOD1 is located primarily in the cytoplasm, SOD2 in the mitochondria and SOD3 is extracellular.</fparagraph><answer_start>Superoxide dismutase 1, or SOD1, is located in the cytoplasm, the material within a eukaryotic cell enclosed by the cell membrane except for the cell nucleus.<answer_end><plan_start>different types of superoxide dismutase (SOD1, SOD2, SOD3) found in the body<plan_end><paragraph>Reactive oxygen species nearly all cells exposed to oxygen. In mammals and most chordates, three forms of superoxide dismutase are present. SOD1 is located primarily in the cytoplasm, SOD2 in the mitochondria and SOD3 is extracellular. The first is a dimer (consists of two units), while the others are tetramers (four subunits). SOD1 and SOD3 contain copper and zinc ions, while SOD2 has a manganese ion in its reactive centre. The genes are located on chromosomes 21, 6, and 4, respectively (21q22.1, 6q25.3 and 4p15.3-p15.1). The SOD-catalysed dismutation of superoxide may be written with the following half-reactions: where M = Cu ("n" =</paragraph><fparagraph>SOD1 is located primarily in the cytoplasm, SOD2 in the mitochondria and SOD3 is extracellular.SOD1 and SOD3 contain copper and zinc ions, while SOD2 has a manganese ion in its reactive centre.</fparagraph><answer_start>SOD2 is found in the mitochondria, an organelle found in most eukaryotic organisms.<answer_end><plan_start>Superoxide dismutase SOD3 found in the body<plan_end><paragraph>SOD3 SOD3 Extracellular superoxide dismutase is an enzyme that in humans is encoded by the "SOD3" gene. This gene encodes a member of the superoxide dismutase (SOD) protein family. SODs are antioxidant enzymes that catalyze the dismutation of two superoxide radicals into hydrogen peroxide and oxygen. The product of this gene is thought to protect the brain, lungs, and other tissues from oxidative stress. The protein is secreted into the extracellular space and forms a glycosylated homotetramer that is anchored to the extracellular matrix (ECM) and cell surfaces through an interaction with heparan sulfate proteoglycan and collagen. A fraction of</paragraph><fparagraph>SOD3 SOD3 Extracellular superoxide dismutase is an enzyme that in humans is encoded by the "SOD3" gene.The protein is secreted into the extracellular space and forms a glycosylated homotetramer that is anchored to the extracellular matrix (ECM) and cell surfaces through an interaction with heparan sulfate proteoglycan and collagen.</fparagraph><answer_start>Finally, SOD3 is extracellular, meaning it occurs in the space outside the plasma membrane of a cell.<answer_end>

Table 11: An Example for ASQA datasets

Input: Which movie came out first Escape to Witch Mountain or Pete's Dragon ?(*HotpotQA*)	
Output: <plan_start>Escape to Witch Mountain release<plan_end><paragraph>Escape to Witch Mountain is a 1975 American fantasy-children's film, adapted from the 1968 science fiction novel of the same name written by Alexander H. Key. The film was produced by Walt Disney Productions, released in March 1975 by Buena Vista Distribution Company and directed by John Hough. </paragraph><fparagraph>Escape to Witch Mountain is a 1975 American fantasy-children's film, adapted from the 1968 science fiction novel of the same name written by Alexander H. Key</fparagraph><answer_start>Escape to Witch Mountain came out first,<answer_end><plan_start>Pete's Dragon release<plan_end><paragraph>Pete's Dragon is a 2016 American fantasy comedy-drama adventure film directed by David Lowery, written by Lowery and Toby Halbrooks, and produced by James Whitaker. The film is a live-action reimagining of Disney's 1977 live-action/animated musical film of the same name written by Malcolm Marmorstein. The film stars Bryce Dallas Howard, Oakes Fegley, Wes Bentley, Karl Urban, Oona Laurence, and Robert Redford. The film tells the story of an orphaned feral boy who befriends a dragon in the Pacific Northwest, and the ensuing repercussions of their discovery by the town's local residents. </paragraph><fparagraph>Pete's Dragon is a 2016 American fantasy comedy-drama adventure film directed by David Lowery, written by Lowery and Toby Halbrooks, and produced by James Whitaker. </fparagraph><answer_start>before Pete's Dragon. <answer_end>[Combine]<answer_start>Escape to Witch Mountain<answer_end>	

Table 12: An Example for HotpotQA datasets