Modeling Uncertainty and Using Post-fusion as Fallback Improves Retrieval Augmented Generation with LLMs

Anonymous EMNLP submission

Abstract

The integration of retrieved passages and large language models (LLMs), such as ChatGPTs, has significantly contributed to improving open-domain question answering. However, there is still a lack of exploration regarding the optimal approach for incorporating retrieved passages into the answer generation process. This paper aims to fill this gap by investigating different methods of combining retrieved passages with LLMs to enhance answer generation. We begin by examining the limitations of a commonly-used concatenation approach. Surprisingly, this approach often results in generating "unknown" outputs, even when the correct document is among the top-k retrieved passages. To address this issue, we explore four alternative strategies for integrating the retrieved passages with the LLMs. These strategies include two single-round methods that utilize chain-of-thought reasoning and two multi-round strategies that incorporate feedback loops. Through comprehensive analyses and experiments, we provide insightful observations on how to effectively leverage retrieved passages to enhance the answer generation capability of LLMs. On three open-domain question answering datesets, NQ, TriviaQA and SQuAD, our multi-round approaches outperform traditional concatenation approach, achieving over a 10% improvement in answer EM.

1 Introduction

Large Language Models (LLMs), such as GPTs (Brown et al., 2020; Bubeck et al., 2023), have found extensive applications, but often struggle with limited knowledge representation, resulting in inaccuracies and insufficient specificity in open-domain question answering. To overcome these limitations, the integration of retrieval-based techniques (Izacard et al., 2022; Borgeaud et al., 2022) has emerged as a promising solution. By incorporating relevant passages during the answer generation, LLMs can leverage external information to provide more accurate and detailed responses. Nevertheless, effective strategies for incorporating retrieved passages into the LLMs remains a challenging and relatively understudied area.

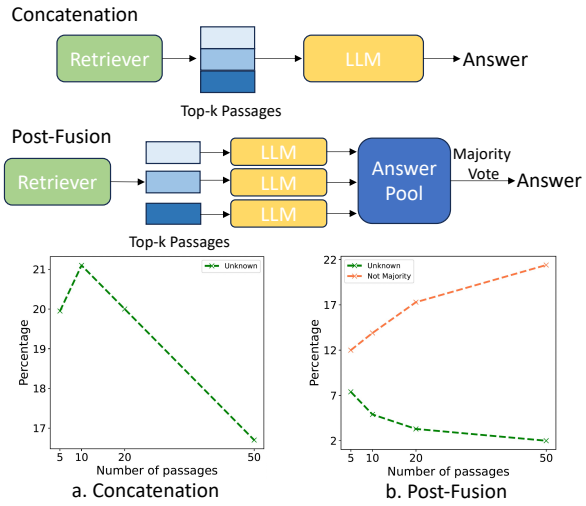

Figure 1: **Top**: Illustration of Concatenation v.s. Post-Fusion strategies. **Bottom-a**: percentage of unknown responses using the Concatenation strategy. **Bottom-b**: by varying the number of retrieved passages, (green) percentage of unknown responses, and (red) error rate by majority voting (when the correct answer is in the answer pool, the majority selects a wrong answer).

Our analysis (Fig. 1), conducted under the oracle setting where one of the top-k retrieved passages contains the answer, reveals that a simple concatenation of the passages into LLMs often leads to "unknown" responses — instances where the provided context fails to answer the question — accounting for about 20% of all responses. An alternative method, where the passages are individually provided as input to LLMs and the majority vote determines the final answer, reduces the rate of "unknown" generation to 2-7% depending on the number of passages. However, this method introduces a new challenge: the correct answer does not align with the majority vote in the answer pool. Particularly, when more passages are incorporated

from 5 to 50, the error rate of the majority vote increases from 12% to 22%. Thus, both of the methods have their own weaknesses and more suitable approaches for the integration of retrieved passages and LLMs remain to be investigated.

Transformer-based LLMs have shown the capability to utilize attention mechanisms (Vaswani et al., 2017) for discovering token-level relevance. However, they may not always attend to the relevant parts within the context, leading to a potential oversight of important information present in the retrieved passages (Clark et al., 2019; Zhao et al., 2019). This challenge becomes more pronounced when dealing with extensive corpora like Wikipedia, which contains over 21 million passages, making it a formidable task to identify the most relevant passages for a question. Furthermore, retrieved passages that are closely related to the question's topic can act as distractors, potentially misleading the model (Asai et al., 2019). If the model mistakenly directs its attention towards these distractor passages, it can introduce noise that negatively impacts the answer prediction process.

In this paper, we explore the integration of retrieved passages with LLMs like ChatGPTs to enhance their ability to generate correct answers. In particular, we examine situations where the retrieved passages contain the correct answer, yet the model fails to generate the correct response, indicating areas for improvement. Initially, we focus on two chain-of-thought (CoT) (Wei et al., 2022; Wang et al., 2022; Trivedi et al., 2022a) strategies that incorporate in-context learning. We introduce a pruning strategy and a summarization strategy for the retrieved passages to guide the answer generation process of the LLMs.

Subsequently, we investigate two multi-round methods with feedback: **Post-Fusion as the Fallback**: In the initial round, this method employs the Concatenation approach to generate an answer. If the LLM generates "unknown" responses with the inputs, it proceeds to use Post-Fusion in the second round, generating candidate answers. The final answer is chosen via majority vote. **Concatenation as the Distiller**: This approach starts by leveraging Post-Fusion to produce a pool of potential answers and to identify relevant passages. In the subsequent round, only the unfiltered passage is concatenated with the question and answer candidates from the first round. This consolidated input is then fed into the LLM to derive the final answer.

Through extensive experiments on three single-hop open-domain question-answering datasets, we showcase the enhanced performance of our proposed methods, achieved with a minimal additional resource cost. Our findings provide a foundation for the development of more advanced retrieval-integration methods aimed at further enhancing the capabilities of these models.

2 Problem Setup

This study focuses on the question answering task under the open-domain setting. It remains a open problem to retrieve the most relevant context for question answering. Therefore, a common practice is to include multiple top ranked passages, which serves as the supplementary context for the LLMs. The number of supplementary passages, denoted as k, can vary based on the desired input length M of the LLM. Typically, k can be set to 5, 10, or 20, ensuring that the total length of k passages, each having a maximum length of L, remains within the maximum input length M of the LLM (i.e., $k * L < M$). By incorporating these supplementary passages, the LLM is provided with a more comprehensive and informative context, which has the potential to enhance its accuracy.

3 Methods

We adopt a two-stage pipeline for open-domain QA. It consists of two black-box components, a retriever and a LLM such as ChatGPT and LLama2 (Touvron et al., 2023). We aim to methodically investigate the optimal methods for transferring the top-k retrieval results to the LLMs for generating factoid answers. Our investigation begins with a focus on various **single-round** strategies, wherein the retrieved passages are directly fed into the LLMs. Subsequently, we delve into several **multi-round** approaches, involving the initial supply of retrieved passages to the LLMs, gathering feedback, and then modifying the interaction process with the LLMs based on that feedback.

3.1 Definition of Unknown Output

LLMs are not universally capable. Their effectiveness relies on being trained on relevant data, storing essential knowledge within their weights. When an LLM cannot provide an answer directly, a common strategy is to use retrieval to fetch pertinent context. However, there may be instances where the model discerns that the retrieved context is insufficient for

a response. In such cases, the LLM might produce outputs like "The provided input does not contain the context to answer the question." We interpret this behavior as the LLM's self-awareness of its inability to confidently produce an answer based on the top-k retrieved passages. To standardize the model's response in these situations and prevent varied output formats, we prompt the model to generate "unknown" when it believes the given context is inadequate for an answer. To be specific, we add the following sentence in the prompt: "*If don't know the answer, just say Unknown.*"

3.2 Single-Round Approaches

In this section, we explore single-round strategies where retrieved passages are directly sent to the LLM. We first examine a zero-shot approach, providing only the task definition and desired output format, without demo examples. Then, we study a one-shot strategy, utilizing a single demo example to guide the LLM's answer generation.

3.2.1 Zero-shot Prompt

Our first line of investigation pertains to a zero-shot setting. In this setting, we only provide the task definition and the desired answer format as the prompt, excluding any demonstration examples that elucidate how to generate an answer from the question and the Top-k passages.

Concatenation Prompt. We begin our exploration with a straightforward and commonly used method that involves concatenating the question and the retrieved passages. These passages are arranged in the order they were retrieved and combined into a single text string. This composite text is then fed into the language model to generate the final answer, which can be represented by the below equation:

$$a = \text{LLM}(q, p_1, p_2, ..., p_k) \tag{1}$$

From our experimental results, we observe that this approach can potentially lead to "unknown" output, even when one of the retrieved passages contains the ideal context necessary to answer the question. This stems from the LLM possibly becoming confused due to the complexity or abundance of input, subsequently generating an unsatisfactory response.

Post-Fusion Prompt. We also explored an alternative approach where each of the Top-k retrieved passages is independently fed to the LLM. After generating an answer for every passage, the collective responses form an answer pool. A majority voting mechanism is then applied to this pool to determine the final answer, which can be denoted by the following equation:

$$\begin{aligned} a_1 =& \text{LLM}(q, p_1), \cdots, a_k = \text{LLM}(q, p_k) \\ & \text{majority} = \arg\max_i a_i \end{aligned} \tag{2}$$

Our experimental findings suggest that while this approach can decrease the likelihood of indeterminate output, it presents a distinct challenge. Specifically, the correct or "gold" answer may indeed be presented within the generated answer pool, but it might not be the majority answer, thus resulting in an incorrect final response.

3.2.2 Few-shot Prompt

We introduce two distinct prompts, with one-shot example, to guide the LLMs in fusing answers from potentially relevant passages. Examples of these two prompt types are provided in Fig. 8 and 9 in the Appendix A, respectively.

Given the significant enhancements chain-of-thought brings to multi-hop question answering, we aim to adapt this approach for single-hop retrieval-augmented generation. Our method uses demonstrative examples to guide answer generation strategies. We employ two techniques for this: One approach involves pruning irrelevant passages and using the few remaining relevant ones for answer generation. The other one is to initially identify the relevant information and then summarize the relevant information like chain of thought and generate the final answer.

Pruning Prompt. This prompt requires the LLM to effectively identify answerable passages through a process of selective elimination. As a result, The demonstration involves differentiating irrelevant passages from the ones that can provide an answer, and subsequently generating the final response based on the few relevant passages.

Summary Prompt. Summarization represents a strategy that extracts the central information from the Top-k passages. Based on this synthesized summary, the LLM can produce the final answer. We posit that summarization could serve as a guiding mechanism for the LLM to more effectively respond to the question. To illustrate this, we provide a demonstration example that exhibits how the model selects useful information from the passage before delivering the final response.

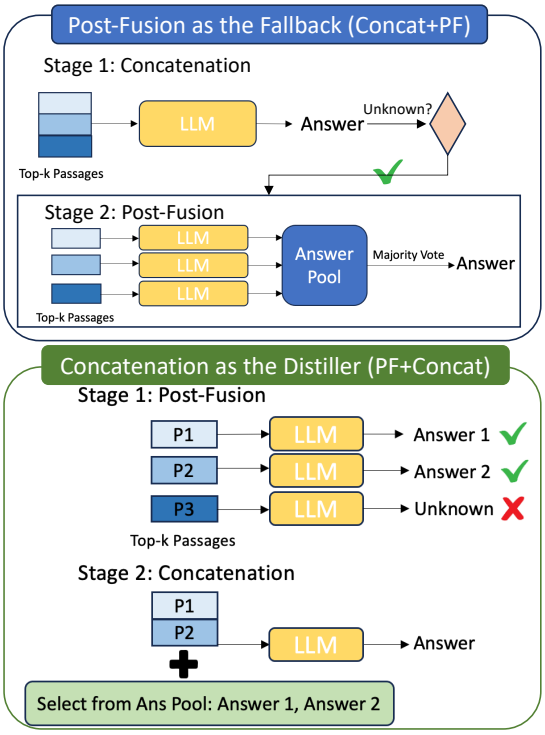

Figure 2: Diagram of Post-Fusion as the Fallback on top and Concatenation as the Distiller at bottom.

3.3 Multi-Round Approaches

In our exploration of multi-round strategies, we first provide the retrieved passages to the LLM. Based on the initial feedback received either "unknown" or a list of candidate answers, we adjust our interaction process with the LLM accordingly.

Post-Fusion as the Fallback (Concat+PF). Initially, we employ the concatenation method as illustrated in upper box of Fig. 2 to obtain an answer predicted by the LLM. If the LLM determines that the input passages are unable to provide an answer to the question (i.e., "unknown" responses), we then proceed to the second round where we utilize the Post-Fusion approach to produce an answer pool. Finally, we employ a majority vote to select the final answer.

Concatenation as the Distiller (PF+Concat). To begin with, we leverage the Post-Fusion strategy to curate a pool of potential answers shown in lower box of Fig. 2. Instead of performing a majority vote, a passage selection process (Lewis et al., 2020) is adopted to discard passages that yield an "unknown" output by the LLM. In the second round, the LLM is prompted with the concatenation of the unfiltered passages, along with the question and answer candidates generated from the first round. The purpose is to guide the LLM in effectively extracting (distilling) the correct answer from the pool of candidates.

4 Experiments

Evaluation Benchmarks. We conduct evaluations on multiple datasets of open-domain question answering to assess the performance of the proposed integration approaches.

The datasets used include Natural Questions (NQ) (Kwiatkowski et al., 2019), TriviaQA (Trivedi et al., 2022b), and SQuAD-Open (Ho et al., 2020) are all datasets designed for training and evaluating single-hop question answering models. NQ is sourced from Google Search queries and their corresponding Wikipedia answers. TriviaQA offers a broader domain with trivia questions and their answers derived from web and Wikipedia sources. Conversely, SQuAD-Open is a variant of the original SQuAD dataset that requires the model to extract answers from open-domain Wikipedia content, without any pre-specified passage.

Evaluation Metrics We adhere to traditional QA dataset evaluation methods (Yang et al., 2018; Ho et al., 2020), contrasting with the recent LLM evaluations on QA tasks detailed in (Liu et al., 2023), which assess whether the generated answer includes the ground truth. Importantly, our evaluation criteria are more rigorous than these recent LLM evaluations (Liu et al., 2023), given that we mandate the LLM to adhere strictly to the given prompt in generating an entity-specific answer. In detail, predicted answers are evaluated with the standard answer exact match (EM) and F1 metric (Rajpurkar et al., 2016; Liu et al., 2022). A generated response is considered correct if, after normalization, it matches any candidate in a list of acceptable answers. The normalization process entails converting the text to lowercase and omitting articles, punctuation, and redundant whitespaces.

We also evaluate the percentage of "unknown" responses (%Unk) which gauges the proportion of times the LLM indicates it cannot answer based on the given input. Additionally, we measure the error rate through majority vote (%NM), representing instances where the correct answer is within the generated answer list but isn't the majority selection.

Dataset Filter To mitigate the influence of specific training datasets on the LLM (Aiyappa et al., 2023), we initially prompt the LLM to answer questions without any provided context. This process

	NQ				TriviaQA				SQuAD			
	EM	F1	%Unk	%NM	EM	F1	%Unk	%NM	EM	F1	%Unk	%NM
With gold passage												
`LLama2`												
Concatenation	26.9	36.9	12.9%	-	38.5	44.9	8.3%	-	37.0	39.3	10.8%	-
Post-Fusion	27.5	38.6	2.8%	27.8%	38.8	45.2	4.4%	19.2%	38.3	42.3	6.8%	8.9%
Pruning Prompt	27.8	37.8	10.9%	-	39.3	45.9	7.8%	-	35.3	41.7	8.4%	-
Summary Prompt	28.1	37.9	9.8%	-	39.2	45.2	7.5%	-	38.5	42.6	7.9%	-
Concat + PF	**30.3**	**40.5**	**1.7%**	3.8%	40.4	46.0	**0.8%**	2.6%	**41.5**	**45.1**	**3.6%**	6.3%
PF + Concat	29.6	39.8	2.7%	**2.3%**	**40.7**	**46.6**	3.9%	**1.5%**	40.2	44.3	4.8%	**5.6%**
`ChatGPT`												
Concatenation	38.1	45.4	19.9%	-	51.6	57.9	18.1%	-	53.1	64.9	13.6%	-
Post-Fusion	40.1	50.4	7.4%	12.0%	51.4	57.3	9.1%	10.2%	57.1	71.2	2.1%	4.3%
Pruning Prompt	39.0	50.5	6.9%	-	52.7	59.5	8.1%	-	47.7	62.6	6.7%	-
Summary Prompt	40.5	53.3	**5.1%**	-	51.6	60.1	6.4%	-	50.4	67.0	4.7%	-
Concat + PF	42.9	53.9	6.5%	3.8%	**55.9**	**62.8**	7.5%	4.3%	60.6	74.0	**1.7%**	2.2%
PF + Concat	**43.2**	**54.5**	5.4%	**3.6%**	54.0	61.7	**6.2%**	**3.1%**	**63.9**	**76.9**	2.1%	**2.0%**
`GPT4`												
Concatenation	41.9	52.9	14.9%	-	54.1	61.8	12.7%	-	57.0	63.9	9.8%	-
Post-Fusion	39.7	51.7	**5.5%**	13.4%	55.0	63.2	8.9%	11.8%	58.2	64.5	3.5%	6.7%
Pruning Prompt	41.2	52.3	6.2%	-	55.2	62.8	**4.5%**	-	57.2	63.1	7.5%	-
Summary Prompt	40.6	52.6	7.4%	-	54.8	62.5	5.9%	-	57.8	62.7	6.5%	-
Concat + PF	**44.3**	**55.1**	6.4%	**2.1%**	**58.3**	**67.4**	7.1%	**3.2%**	**66.2**	**78.4**	**3.8%**	**1.1%**
PF + Concat	43.8	54.6	7.3%	4.2%	57.8	66.2	9.5%	7.3%	65.3	77.9	4.2%	3.6%

Table 1: Exact match (EM) and F1 scores on filtered DEV split of the NQ, TriviaQA and SQuAD using Top-5 passages under with gold passage setting. %Unk denotes the percentage of Unknown responses. %NM denotes the error rate by majority vote. **Concat** refers to the Concatenation strategy and **PF** refers to Post-Fusion strategy.

enables us to filter out questions that the LLM can accurately answer independently, thereby eliminating the need for additional external contextual information. The remaining questions, which the LLM couldn't answer independently, are the focus of our study. This filtering ensures our evaluation stringently reflects the LLM's ability to utilize external context from retrieved passages.

We use the development set of NQ, TriviaQA, and SQuAD, initially containing 5,892, 6,760, 5,928 questions, respectively. After removing questions that can be answered without context, we are left with 3,459 questions in NQ, 1,259 in TriviaQA, and 3,448 in SQuAD.

Retriever and LLM model. We use the Wikipedia dump from Dec. 20, 2018 for NQ and TriviaQA and the dump from Dec. 21, 2016 for SQuAD. We apply preprocessing steps following Chen et al. (2017); Karpukhin et al. (2020); Liu et al. (2021), which involve generating non-overlapping passages of 100 words each. Similar to (Izacard and Grave, 2021), passages are retrieved with DPR (Karpukhin et al., 2020) for NQ and TriviaQA and with BM25 (Robertson et al., 1995) for SQuAD. We consider two different settings for this study. The first utilizes the top-k retrieved passages directly (gold passage is not necessarily included). In contrast, the second setting concerns the situation that the gold-standard passage is included in the context. If the gold passage is not within the top-k passages, we randomly insert it into the top-k list.

We use both open and close LLMs. For Llama2 (Touvron et al., 2023), we use the instruction-tuned version `Llama-2-7b-chat-hf` model and apply greedy decoding with the temperature parameter set to 0. For ChatGPT, we use the `gpt-3.5-turbo-16k` model. For GPT4 (OpenAI, 2023), our choice is `gpt-4-0613`.

4.1 Results

The results using the gold passages setting are presented in Table 1, while those without incorporating gold passages are in Table 2. Initially, we obtain the Top-5 retrieved passages, representing the setting without added gold passages. If these passages don't contain the answer, we randomly integrate the gold passage among the Top-5 candidate passages, corresponding to the setting with gold passages.

Table 1 reveals that among the single-round zero-shot methods, Post-Fusion consistently surpasses the traditional concatenation approach in both EM and F1 metrics across all three bench-

	NQ				TriviaQA				SQuAD			
	EM	F1	%Unk	%NM	EM	F1	%Unk	%NM	EM	F1	%Unk	%NM
Supervised	40.9	-	-	-	55.2	-	-	-	35.8	-	-	-
Without gold passage												
`LLama2`												
Concatenation	24.6	34.6	18.2%	-	35.8	40.9	14.6%	-	20.1	28.9	21.8%	-
Post-Fusion	24.9	36.3	13.8%	15.3%	35.9	43.8	10.5%	14.5%	21.5	29.5	16.2%	18.3%
Pruning Prompt	25.7	35.4	12.7%	-	36.2	43.9	9.8%	-	23.5	30.4	10.4%	-
Summary Prompt	26.3	35.7	10.3%	-	36.2	42.0	8.5%	-	23.8	30.2	10.9%	-
Concat + PF	**28.0**	**38.9**	**3.2%**	**3.6%**	37.7	43.2	**4.2%**	3.5%	**26.5**	34.9	**3.2%**	2.6%
PF + Concat	27.9	38.5	8.7%	4.8%	**38.2**	**43.6**	8.9%	**2.8%**	24.2	**35.8**	12.8%	**2.3%**
`ChatGPT`												
Concatenation	34.5	43.8	23.1%	-	49.3	55.5	19.9%	-	28.1	34.8	28.5%	-
Post-Fusion	38.3	48.3	10.1%	9.0%	49.7	55.7	10.7%	7.4%	32.1	40.3	13.9%	12.3%
Pruning Prompt	36.2	46.3	9.1%	-	49.3	56.5	9.5%	-	36.1	40.6	12.7%	-
Summary Prompt	36.3	48.4	**8.6%**	-	48.3	56.5	**7.7%**	-	34.1	40.0	13.7%	-
Concat + PF	**39.9**	49.7	9.3%	5.3%	**52.7**	**59.5**	9.1%	**2.8%**	**40.1**	**43.8**	**5.7%**	**2.3%**
PF + Concat	38.9	**50.1**	9.1%	**4.3%**	50.5	57.7	6.7%	3.2%	38.5	41.2	9.9%	5.4%
`GPT4`												
Concatenation	36.9	50.6	18.9%	-	51.3	60.7	16.7%	-	29.7	30.9	25.8%	-
Post-Fusion	37.7	49.7	6.5%	9.9%	51.5	59.0	13.2%	8.9%	33.1	37.8	12.8%	12.5%
Pruning Prompt	38.3	48.4	9.2%	-	51.2	58.2	12.5%	-	32.7	39.8	13.6%	-
Summary Prompt	38.5	49.6	8.3%	-	50.8	58.5	13.9%	-	35.9	39.2	12.5%	-
Concat + PF	**41.5**	**52.1**	**5.4%**	**3.1%**	**55.7**	**63.7**	**8.1%**	**3.8%**	41.8	44.7	**5.6%**	**3.2%**
PF + Concat	40.6	51.6	6.9%	9.2%	54.3	62.8	12.5%	6.4%	**42.1**	**44.9**	9.7%	8.4%

Table 2: Exact match (EM) and F1 scores on filtered DEV split of the NQ, TriviaQA and SQuAD using Top-5 passages on without adding gold passage setting. %Unk denotes the percentage of Unknown responses. %NM denotes the error rate by majority vote. **Concat** refers to the Concatenation strategy and **PF** refers to Post-Fusion strategy.

marks. This indicates that the model may become distracted when faced with a combination of relevant passages. Compared to zero-shot and few-shot approaches, both Pruning Prompt and Summary Prompt show a marked enhancement over the concatenation method, though the margin of improvement is modest. The use of the CoT, which elicits a potential reasoning process, can guide the model in attending to relevant passages. However, this approach does not greatly enhance single-hop question answering as compared to prior multi-hop reasoning studies (Wei et al., 2022; Trivedi et al., 2022a).

Compared to single-round methods, multi-round strategies consistently deliver superior performance, showcasing significant improvements. For instance, on the NQ dataset, Concat + PF exceeds the Concatenation method by over 10% on average across three distinct LLMs. It suggests the efficacy of integrating model uncertainty as feedback. Among the multi-round approaches, Concat + PF demonstrates better performance compared to PF + Concat on most of cases. Comparing PF + Concat with Post-Fusion, it is evident that PF + Concat, leveraging LLM to select the best answer from a candidate pool, outperforms the majority vote approach.

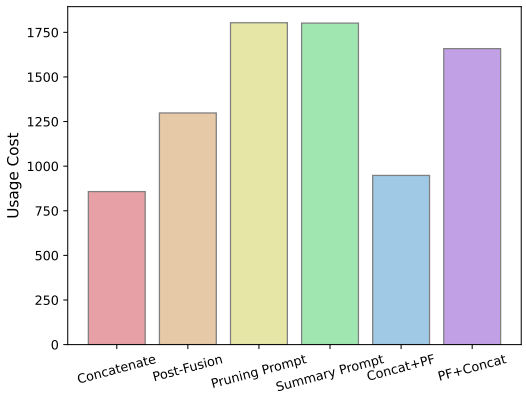

Figure 3: The token usage of different approaches using top-5 passages.

In the realm of open-domain question-answering, as evidenced by Table 2, the performance metrics (EM and F1) under settings without the addition of a gold passage are comparatively lower. This is primarily attributed to the reduced recall of Top-k retrieval, resulting in a higher propensity to generate "unknown" responses. Notably, our proposed multi-round methodologies, when leveraging GPT4 as the LLM, deliver performance figures that are on par with supervised outcomes.

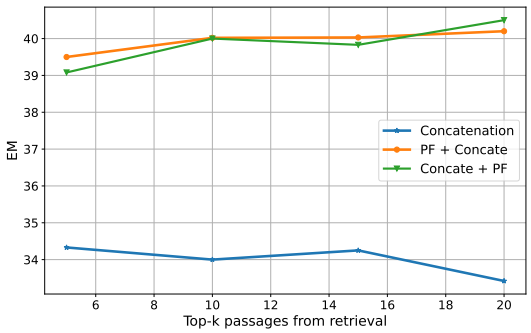

Figure 4: The impact on the position of gold passage on Combination method.

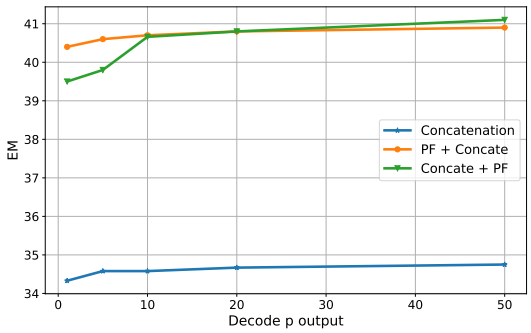

Figure 5: The impact on the position of gold passage on Combination method.

4.2 Usage Analysis

Striking a balance between enhancing the quality of generated answers and optimizing resource utilization is essential. As depicted in Figure 3, different methodologies vary in their token usage. The Concatenate method is the most resource-efficient, whereas the Concat + PF method, albeit being the second most efficient, has an additional 90.5 tokens on average when compared to Concatenate. Given the significant performance boost of Concat + PF over Concatenate (a 15.6% increase in EM as presented in Table 2), we advocate for the adoption of Concat + PF. This offers a more efficient means of integrating retrieved passages with LLMs.

4.3 Effect of different Top-k passages from the retriever

Figure 4 showcases open-domain QA results using the Top-k retrieved passages on NQ dataset. As k increases, we observe a corresponding increase in retrieval recall. Our multi-stage methods, Concat + PF and PF + Concat, both benefit from increasing k values, showing enhancements of 1.5 and 0.7 points, respectively, when moving from Top 5 to 20. In contrast, the conventional concatenation method experiences a 0.8 EM performance decline from Top 5 to 20. This suggests that the concatenation prompt can become counterproductive with the inclusion of more passages, potentially because it struggles to identify the correct passage and gets distraction by incorrect ones. However, our multi-stage approaches remain undeterred with the addition of passages, demonstrating greater robustness.

4.4 Effect of different Decoding Strategies

Instead of the traditional greedy decoding strategy, a newer method known as self-consistency (Wang et al., 2022) has been introduced and employed in the chain-of-thought prompting (Wei et al., 2022). This method begins by sampling from the language model's decoder to produce a diverse set of answers. The optimal answer is then obtained by marginalizing the samples' reasoning paths.

For the concatenation prompt, we opt for temperature sampling (Ackley et al., 1985; Ficler and Goldberg, 2017) as our decoding strategy, yielding p outputs, rather than generating a singular answer via greedy decoding as detailed in section 4.1. In the case of the post-fusion prompt, each passage employs a sampling decoding strategy, generating p outputs for every k passages. This results in a total of $p \times k$ outputs. It's important to distinguish between post-fusion prompts and self-consistency. The former pertains to using different inputs, while the latter is about the decoding sampling strategy.

Figure 5 presents an ablation of results with a temperature of 0.7 and varying values of p in Top-p sampling on ChatGPT, using the Top-5 retrieved passages from the NQ dataset. The data suggests that small sampling outputs, ranging from 1 to 10, significantly enhance performance. However, as p increases from 10 to 50, the degree of improvement diminishes. And Concate + PF approach could benefit more from the increase of p.

4.5 Effect of the order of the gold passage

In this section, we aim to assess how the placement of the gold passage within the Top-k passages influences the ability of the LLM to generate accurate answers. We examine three different placements: (1) consistently positioning the gold passage at the start of the Top-k passage list; (2) consistently placing the gold passage at the end of the Top-k passage list; (3) maintaining the original sequence produced by the retrieval model.

As the results depicted in Fig. 6, it is evident that the placement of the gold passage significantly

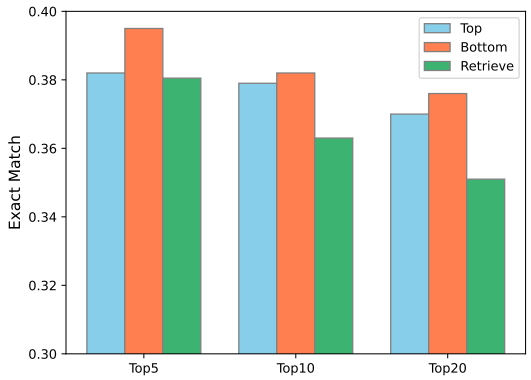

Figure 6: The impact on the position of gold passage on Combination method.

affects the quality of the generated answers. Consistently placing the gold passage in the same position tends to improve performance compared to using the retrieval order. Among the constant placement options, positioning the gold passage at the bottom tends to yield better results than placing it at the top. This outcome might be tied to our prompt design, where we present the Top-k passages first, followed by the question. Consequently, keeping the gold passage closer to the question seems to enhance performance to the greatest extent. Moreover, this observation is aligned with the (Liu et al., 2023), where they find that a distinctive U-shaped performance, as performance peaks when key information is at the start or end of the input, but drops significantly for mid-context information.

5 Related Work

The recent proliferation of LLM-powered applications, such as ChatGPT/GPT4 (OpenAI, 2023), Bing Chat, and CoPilot, has highlighted both the impressive performance and certain limitations of LLMs. These limitations include a high compute and data demand, making it a challenge to continually update LLMs both efficiently and effectively (Scialom et al., 2022). LLMs also tend to generate plausible yet non-factual texts, a phenomenon known as "hallucination" (OpenAI, 2023). In response to these issues, the field is witnessing a trend towards augmenting LLMs with specialized tools (Schick et al., 2023; Paranjape et al., 2023), such as code interpreters (Zhang et al., 2021; Gao et al., 2023; Shao et al., 2023) or search engines (Park and Ryu, 2023). The goal is to delegate specific tasks to more proficient systems or to enrich the LLMs' input context with more pertinent information.

Augmentation of language models with pertinent data retrieved from diverse knowledge bases has demonstrated its effectiveness in enhancing open-domain question answering performance (Lazaridou et al., 2022; Izacard et al., 2022; Chen et al., 2023). The process typically involves using the input query to (1) command a retriever to fetch a document set (essentially, token sequences) from a corpus, after which (2) the language model integrates these retrieved documents as supplemental information, guiding the final prediction.

The interleaving between the retriever and LLM could be considered a reciprocal process. Various studies have been conducted on generation-augmented retrieval (GAR), which involves revising or supplementing queries with generated background information to enhance the retrieval of relevant content. Well-known examples of this approach include GAR (Mao et al., 2021) and HyDE (Gao et al., 2022). With regard to complex multi-step reasoning questions, work involving LLMs often necessitates the retrieval of segmented knowledge (Trivedi et al., 2022a; Khattab et al., 2022). This chain-of-thought reasoning process (Wei et al., 2022; Jiang et al., 2023) is followed by conducting partial reasoning to generate the next question, then retrieving further information based on the outcome of that partially formed next question, and repeating this cycle as needed (Yao et al., 2022; Press et al., 2022).

Our work primarily focuses on a specific scope: once the output from the retriever is determined, we aim to identify the most effective method of inputting this data into LLMs for answer generation.

6 Conclusion

In this study, we identified two key challenges associated with integrating LLMs and retrieved passages: the occurrence of "unknown" responses when feeding LLMs with concatenated passages and the erroneous majority when using the Post-Fusion approach. To overcome these challenges, we proposed four improved approaches, including two CoT-related strategies and two multi-round methods incorporating LLM's feedback. Through our experimental results and token usage analysis, we observed that it is advantageous to first employ a concatenation strategy to generate an answer. In the case of an "unknown" response, we recommend transitioning to the Post-Fusion approach to obtain the final answer through a majority vote.

Limitations

Our evaluation is primarily constrained to three open-domain QA datasets to align better with the supervised state-of-the-art approach cited in (Izacard and Grave, 2021). To ensure the broader applicability and robustness of our findings, it's essential to evaluate the proposed methods on other benchmarks, including MS MARCO and WebQuestions datasets (Nguyen et al., 2016; Berant et al., 2013).

Currently, our evaluation focuses predominantly on textual QA. While the proposed approach seems generalizable to other modalities like tables (Pasupat and Liang, 2015; Zhu et al., 2021) and knowledge bases (Berant et al., 2013; Bao et al., 2016), we have yet to empirically test and validate this claim. Future studies could delve into exploring its effectiveness on diverse modalities like UniK QA (Oguz et al., 2022).

We haven't thoroughly evaluated how our approach scales with larger datasets or more complex queries (Trivedi et al., 2022b). This could be an avenue of exploration, as scalability is vital for real-world applications.

References

David H Ackley, Geoffrey E Hinton, and Terrence J Sejnowski. 1985. A learning algorithm for boltzmann machines. *Cognitive science*, 9(1):147–169.

Rachith Aiyappa, Jisun An, Haewoon Kwak, and Yong-Yeol Ahn. 2023. Can we trust the evaluation on chatgpt? *arXiv preprint arXiv:2303.12767*.

Akari Asai, Kazuma Hashimoto, Hannaneh Hajishirzi, Richard Socher, and Caiming Xiong. 2019. Learning to retrieve reasoning paths over wikipedia graph for question answering. In *International Conference on Learning Representations*.

Junwei Bao, Nan Duan, Zhao Yan, Ming Zhou, and Tiejun Zhao. 2016. Constraint-based question answering with knowledge graph. In *Proceedings of COLING 2016, the 26th international conference on computational linguistics: technical papers*, pages 2503–2514.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on freebase from question-answer pairs. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1533–1544.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. 2022. Improving language models by retrieving from trillions of tokens. In *International conference on machine learning*, pages 2206–2240. PMLR.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879.

Wenhu Chen, Pat Verga, Michiel De Jong, John Wieting, and William Cohen. 2023. Augmenting pre-trained language models with qa-memory for open-domain question answering. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1589–1602.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2019. Electra: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*.

Jessica Ficler and Yoav Goldberg. 2017. Controlling linguistic style aspects in neural language generation. *EMNLP 2017*, page 94.

Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2022. Precise zero-shot dense retrieval without relevance labels. *arXiv preprint arXiv:2212.10496*.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Pal: Program-aided language models. In *International Conference on Machine Learning*, pages 10764–10799. PMLR.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6609–6625.

Gautier Izacard and Edouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. In *EACL 2021-16th Conference of the European Chapter of the Association for Computational Linguistics*, pages 874–880. Association for Computational Linguistics.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2022. Few-shot learning with retrieval augmented language models. *arXiv preprint arXiv:2208.03299*.

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

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781.

Omar Khattab, Keshav Santhanam, Xiang Lisa Li, David Hall, Percy Liang, Christopher Potts, and Matei Zaharia. 2022. Demonstrate-search-predict: Composing retrieval and language models for knowledge-intensive nlp. *arXiv preprint arXiv:2212.14024*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Angeliki Lazaridou, Elena Gribovskaya, Wojciech Stokowiec, and Nikolai Grigorev. 2022. Internet-augmented language models through few-shot prompting for open-domain question answering. *arXiv preprint arXiv:2203.05115*.

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.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023. Lost in the middle: How language models use long contexts. *arXiv preprint arXiv:2307.03172*.

Ye Liu, Kazuma Hashimoto, Yingbo Zhou, Semih Yavuz, Caiming Xiong, and S Yu Philip. 2021. Dense hierarchical retrieval for open-domain question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 188–200.

Ye Liu, Semih Yavuz, Rui Meng, Dragomir Radev, Caiming Xiong, and Yingbo Zhou. 2022. Uni-parser: Unified semantic parser for question answering on knowledge base and database. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8858–8869.

Yuning Mao, Pengcheng He, Xiaodong Liu, Yelong Shen, Jianfeng Gao, Jiawei Han, and Weizhu Chen. 2021. Generation-augmented retrieval for open-domain 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 4089–4100.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. Ms marco: A human generated machine reading comprehension dataset. *choice*, 2640:660.

Barlas Oguz, Xilun Chen, Vladimir Karpukhin, Stan Peshterliev, Dmytro Okhonko, Michael Schlichtkrull, Sonal Gupta, Yashar Mehdad, and Scott Yih. 2022. Unik-qa: Unified representations of structured and unstructured knowledge for open-domain question answering. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1535–1546.

OpenAI. 2023. Gpt-4 technical report.

Bhargavi Paranjape, Scott Lundberg, Sameer Singh, Hannaneh Hajishirzi, Luke Zettlemoyer, and Marco Tulio Ribeiro. 2023. Art: Automatic multi-step reasoning and tool-use for large language models. *arXiv preprint arXiv:2303.09014*.

Hyun Jin Park and Changwan Ryu. 2023. Query augmentation using search engine results to improve answers generated by large language models.

Panupong Pasupat and Percy Liang. 2015. Compositional semantic parsing on semi-structured tables. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1470–1480.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A Smith, and Mike Lewis. 2022. Measuring and narrowing the compositionality gap in language models. *arXiv e-prints*, pages arXiv–2210.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392.

Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, Mike Gatford, et al. 1995. Okapi at trec-3. *Nist Special Publication Sp*, 109:109.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. *arXiv preprint arXiv:2302.04761*.

Thomas Scialom, Tuhin Chakrabarty, and Smaranda Muresan. 2022. Continual-t0: Progressively instructing 50+ tasks to language models without forgetting. *arXiv preprint arXiv:2205.12393*.

Zhihong Shao, Yeyun Gong, Yelong Shen, Minlie Huang, Nan Duan, and Weizhu Chen. 2023. Synthetic prompting: Generating chain-of-thought demonstrations for large language models. *arXiv preprint arXiv:2302.00618*.

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*.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022a. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. *arXiv preprint arXiv:2212.10509*.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022b. Musique: Multi-hop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed H Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations*.

Jingfeng Zhang, Haiwen Hong, Yin Zhang, Yao Wan, Ye Liu, and Yulei Sui. 2021. Disentangled code representation learning for multiple programming languages. *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*.

Chen Zhao, Chenyan Xiong, Corby Rosset, Xia Song, Paul Bennett, and Saurabh Tiwary. 2019. Transformer-xh: Multi-evidence reasoning with extra hop attention. In *International Conference on Learning Representations*.

Fengbin Zhu, Wenqiang Lei, Youcheng Huang, Chao Wang, Shuo Zhang, Jiancheng Lv, Fuli Feng, and Tat-Seng Chua. 2021. Tat-qa: A question answering benchmark on a hybrid of tabular and textual content in finance. 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 3277–3287.

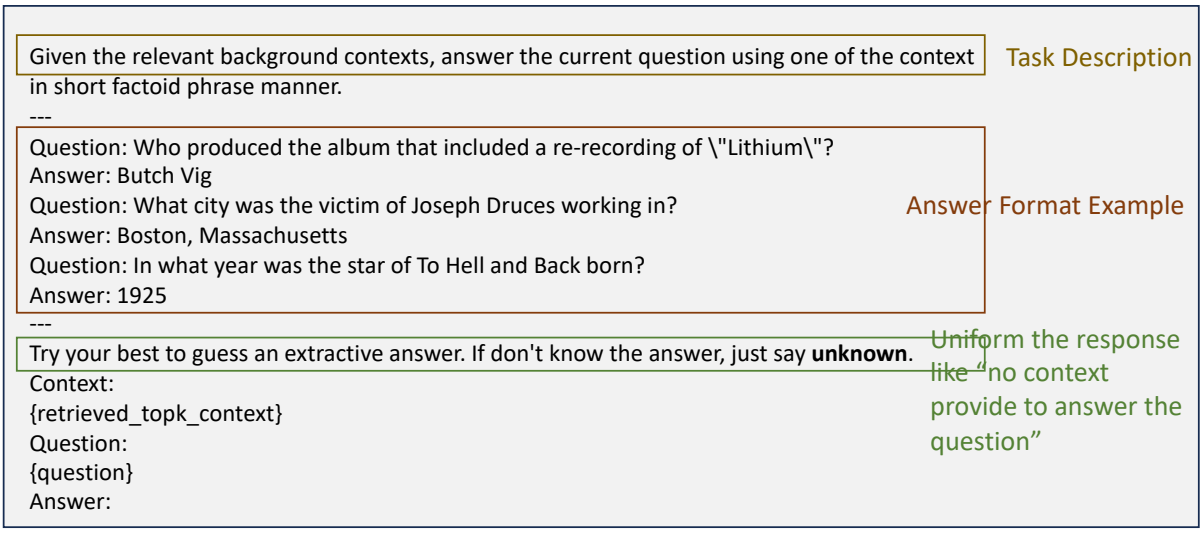

Figure 7: The Prompt used in Concatenation and Post-Fusion.

A Prompt used in Different Approaches

The prompts used in the Concatenation and Post-Fusion approaches are illustrated in Fig. 7. In the Concatenation approach, `retrieved_topk_context` represents the concatenation of the top-k retrieved passages. Conversely, in the Post-Fusion approach, it represents a single passage at a time.

The Pruning Prompt's specific prompt is presented in Fig. 8, while the Summary Prompt's prompt is depicted in Fig. 9.

Answer questions with short factoid answers.

Question: Who produced the album that included a re-recording of \"Lithium\"?
Answer: Butch Vig
Question: What city was the victim of Joseph Druces working in?
Answer: Boston, Massachusetts
Question: In what year was the star of To Hell and Back born?
Answer: 1925

Answer Format Example

Follow the following format.
Context:
sources that may contain relevant content
Question:
the question to be answered
Rationale: Let's think step by step. a step-by-step deduction that identifies the correct response, which will be provided below
Answer: a short factoid answer, often between 1 and 5 words. Make sure generate \"Answer\": in the end!
If don't know the answer, just say **unknown** as answer.

Reasoning and Output Format

Context:
[1] Peter Outerbridge | Peter Outerbridge Peter Outerbridge (born June 30, 1966) is a Canadian actor.....
[2] Except the Dying | 2008. On March 3, 2015, Acorn Media announced a re-release for all three movies, set for May 26, 2015.....
[3] «Saw VI | Saw VI Saw VI is a 2009 American horror film directed by Kevin Greutert from a screenplay written by Patrick Melton and Marcus Dunstan. It is the sixth installment in the \"Saw\" franchise and stars Tobin Bell.....
Question: Which 2009 movie does Peter Outerbridge feature as William Easton?
Rationale: Let's think step by step.
The question is asking for the 2009 movie that Peter Outerbridge was in as William Easton. We can use process of **pruning** to figure this out. Source 1 doesn't contain the information. In source 2, it talks about a made-for-TV movie in 2004. In source 3, it talks about the sixth installment in the \"Saw\" franchise. This must be the movie we are looking for.
Answer:
Saw VI

Elimination Reasoning Demo

Context:
{retrieved_topk_context}
Question:
{question}
Rationale: Let's think step by step.

Figure 8: The Pruning Prompt.

Answer questions with short factoid answers.

Question: Who produced the album that included a re-recording of \"Lithium\"?
Answer: Butch Vig
Question: What city was the victim of Joseph Druces working in?
Answer: Boston, Massachusetts
Question: In what year was the star of To Hell and Back born?
Answer: 1925

Answer Format Example

Follow the following format.
Context:
sources that may contain relevant content
Question:
the question to be answered
Rationale: Let's think step by step. a step-by-step summary that identifies the correct response, which will be provided below
Answer: a short factoid answer, often between 1 and 5 words. Make sure generate \"Answer\": in the end!
If don't know the answer, just say **unknown** as answer.

Reasoning and Output Format

Context:
[1] Peter Outerbridge | Peter Outerbridge Peter Outerbridge (born June 30, 1966) is a Canadian actor.....
[2] Except the Dying | 2008. On March 3, 2015, Acorn Media announced a re-release for all three movies, set for May 26, 2015.....
[3] «Saw VI | Saw VI Saw VI is a 2009 American horror film directed by Kevin Greutert from a screenplay written by Patrick Melton and Marcus Dunstan. It is the sixth installment in the \"Saw\" franchise and stars Tobin Bell.....
Question: Which 2009 movie does Peter Outerbridge feature as William Easton?
Rationale: Let's think step by step.
The question requires information on the 2009 movie that Peter Outerbridge was in as William Easton. Going through the provided sources, we can narrow down our focus to Source 3 and Source 4 that mention \"Saw VI\", a movie released in 2009, in which Peter Outerbridge starred. By **summarizing** these details, the movie from 2009 featuring Peter Outerbridge is \"Saw VI\".
Answer:
Saw VI

Summary Reasoning Demo

Context:
{retrieved_topk_context}
Question:
{question}
Rationale: Let's think step by step.

Figure 9: The Summary Prompt.