Dr3: Ask Large Language Models Not to Give Off-Topic Answers in Open Domain Multi-Hop Question Answering

Yuan Gao[♠,♣], Yiheng Zhu[♣], Yuanbin Cao[♣], Yinzhi Zhou[♣], Zhen Wu[♠,†], Yujie Chen[♣], Shenglan Wu[♣], Haoyuan Hu[♣], Xinyu Dai[♠]

[♠]National Key Laboratory for Novel Software Technology, Nanjing University
[♣]Artificial Intelligence Department, Cainiao Network
gaoy@smail.nju.edu.cn, {wuz, daixinyu}@nju.edu.cn
{zyh171911, lingzun.cyb, yinzhi.zyz, aisling.cyj, shenglan.wsl, haoyuan.huhy}@cainiao.com

Abstract

Open Domain Multi-Hop Question Answering (ODMHQA) plays a crucial role in Natural Language Processing (NLP) by aiming to answer complex questions through multi-step reasoning over retrieved information from external knowledge sources. Recently, Large Language Models (LLMs) have demonstrated remarkable performance in solving ODMHQA owing to their capabilities including planning, reasoning, and utilizing tools. However, LLMs may generate off-topic answers when attempting to solve ODMHQA, namely the generated answers are irrelevant to the original questions. This issue of off-topic answers accounts for approximately one-third of incorrect answers, yet remains underexplored despite its significance. To alleviate this issue, we propose the Discriminate→Re-Compose→Re-Solve→Re-Decompose (Dr3) mechanism. Specifically, the Discriminator leverages the intrinsic capabilities of LLMs to judge whether the generated answers are off-topic. In cases where an off-topic answer is detected, the Corrector performs step-wise revisions along the reversed reasoning chain (Re-Compose→Re-Solve→Re-Decompose) until the final answer becomes on-topic. Experimental results on the HotpotQA and 2WikiMultiHopQA datasets demonstrate that our Dr3 mechanism considerably reduces the occurrence of off-topic answers in ODMHQA by nearly 13%, improving the performance in Exact Match (EM) by nearly 3% compared to the baseline method without the Dr3 mechanism[1].

Keywords: large language models, open domain multi-hop question answering, off-topic answers, prompting

1. Introduction

Open Domain Multi-Hop Question Answering (ODMHQA) is one of the most challenging tasks in Natural Language Processing (NLP) (Mavi et al., 2022). Unlike Reading Comprehension (RC) tasks that provide paired contexts (Rajpurkar et al., 2016), ODMHQA operates in an open-domain setting thereby requiring models to retrieve contexts from external knowledge sources like Wikipedia (Feldman and El-Yaniv, 2019). Unlike single-hop QA where answers can be derived from a single source, ODMHQA exhibits greater complexity as final answers require reasoning over multiple sources in a multi-hop fashion (Zhu et al., 2021). Therefore, ODMHQA is more realistic than basic QA as it involves multi-hop reasoning over the retrieved contexts in an open-domain setting.

Nowadays, Large Language Models (LLMs) have become a *de facto* choice for solving ODMHQA (Wei et al., 2022; Press et al., 2022; Wang et al., 2023b). Among recent works, Yao et al. (2022) proposed the ReAct paradigm in which LLMs are prompted to solve complex problems, inspiring a series of subsequent studies (Hao et al., 2023; Hsieh et al., 2023; Ruan et al., 2023). ReAct prompts LLMs to generate both reasoning traces and actions to interact with the external world in an interleaved manner, outperforming vanilla acting models (Nakano et al., 2021) while being competitive with pure reasoning approaches (Wei et al., 2022).

However, LLMs encounter the issue of generating **off-topic answers** when solving ODMHQA. Specifically, an off-topic answer refers to when the generated answer is irrelevant to the original question. For example, the answer “Barack Obama” is an off-topic answer to the question “In which year was David Beckham’s wife born?”, where an on-topic answer should be a year rather than a name (more examples can be found in Table 1). The process of solving ODMHQA intrinsically involves steps of problem reasoning, task planning, and tool utilization. During these steps, the generation and accumulation of irrelevant information due to inherent hallucinations of LLMs (Zhang et al., 2023; Bang et al., 2023) can result in off-topic answers. In fact, the analysis presented in Section 2.2 reveals that approximately 1/3 of the incorrect answers are identified as off-topic answers, but this issue has not been sufficiently explored so far.

In this paper, to reduce the occurrence of off-topic answers, we propose the Discriminate→Re-Compose→Re-Solve→Re-Decompose (Dr3) mechanism that performs post-hoc judgment and subsequently corrects the reasoning chain through backtracking. Specifically, the Discrimi-

[†]Corresponding Author

[1] Our code and data will be available at `https://github.com/Gy915/Dr3`.

LREC-COLING 2024, pages 5350–5364
20-25 May, 2024.

No.	Question	Gold Answer	Pred Answer	Correct	On-Topic
1	Who is older Danny Green or James Worthy?	James Worthy	James Worthy	✔	✔
2	“Tunak” is a pop love song by an artist born in which year?	1997	1967	✘	✔
3	Which film was released more recently, The Secret Life Of Pets 2 or Love Me Deadly?	The Secret Life Of Pets 2	Lover 3	✘	✘
4	What languages did the son of Sacagawea speak?	French and English	Edinburgh	✘	✘

Table 1: Examples of off-topic answers. In the 1st example, the predicted answer is both correct and on-topic. In the 2nd example, the answer is incorrect but still on-topic. In the 3rd and 4th examples, the answers are not only incorrect but also off-topic.

nator leverages the intrinsic capabilities of LLMs to determine whether the generated answer is off-topic relative to the original question. In cases where an off-topic answer is detected, the Corrector backtracks and revises the reasoning chain (Re-Compose→Re-Solve→Re-Decompose).

Experimental results on the HotpotQA and 2WikiMultiHopQA datasets demonstrate that our Dr3 mechanism considerably improves the performance of LLMs on ODMHQA. Additionally, we conduct dedicated studies to i) investigate the capability of the Discriminator in capturing off-topic answers and its impact on the consequent correctness of ODMHQA, ii) examine the impacts of three individual components of the Corrector (Re-Compose, Re-Solve, and Re-Decompose) on reducing off-topic answers, iii) explore the effect of sub-question numbers on off-topic answers, and iv) investigate the issue of off-topic answers across different types of questions.

To sum up, our contribution is threefold:

1. To the best of our knowledge, we are the first to point out and analyze the issue of off-topic answers in solving ODMHQA using LLMs.
2. We propose the Dr3 mechanism to reduce the occurrence of off-topic answers in solving ODMHQA using LLMs, which contains a Discriminator to detect off-topic answers by LLMs and a Corrector to heuristically correct the off-topic answers.
3. We conduct extensive experiments on the HotpotQA and 2WikiMultiHopQA datasets to demonstrate the effectiveness of our Dr3 mechanism. The results show that Dr3 reduces the occurrence of off-topic answers by nearly 13%, while improving the question answering performance by nearly 3% in Exact Match (EM) compared to ReAct.

2. Preliminary

In this section, we begin by introducing the ReAct approach, which serves as the basic framework for solving ODMHQA using LLMs. Because the reasoning and planning steps are intertwined within the thought component of vanilla ReAct, which hinders identifying and fixing potential issues for ODMHQA, we modify ReAct to establish an equivalent variant called ReAct+ (marginally outperforming the unmodified version), where we explicitly decouple the sub-questions within the reasoning chain for ODMHQA. Subsequently, we elaborate on the issue of off-topic answers that arise when solving ODMHQA using ReAct+. Furthermore, we provide statistics on the error types associated with off-topic answers, offering insights that inspire our proposed method in the next section.

2.1. ReAct(+)

Vanilla ReAct. To solve ODMHQA using LLMs, Yao et al. (2022) proposed ReAct that uses interleaved steps of reasoning and acting. Specifically, ReAct verbally maintains high-level plans for acting (reason to act), while interacting with external environments such as Wikipedia to incorporate additional information into reasoning (act to reason). Formally, ReAct interacts with the environment in N steps before concluding the final answer. At step i, based on the previous step's **observation**, $o_{i-1} \in \mathcal{O}$ (where $\mathcal{O}$ is the set of all possible observations from the environment), ReAct develops a **thought**, $\tau_i \in \mathcal{T}$ (where $\mathcal{T}$ is the set of all possible LLM-generated thoughts), by reasoning over the available information, and planning the next step hence deliver an **action**, $a_i \in \mathcal{A}$ (where $\mathcal{A}$ is the set of optional actions like **search**[query] to retrieve passages or **finish**[answer] to conclude the question), leading to a new observation, $o_i \in \mathcal{O}$, from the environment. Specifically, the action is selected by ReAct's **policy**, $\pi : H_i \rightarrow a_i$, where the **context** $H_i \equiv (o_0, \tau_1, a_1, o_1, \ldots, \tau_{i-1}, a_{i-1}, o_{i-1}, \tau_i)$ represents the interaction history between ReAct and the environment.

Modified ReAct. It is noteworthy that ReAct's reasoning and planning steps are intertwined within its thought component, denoted as τ_i at step i. This entanglement presents challenges in identifying potential issues when solving ODMHQA, which hinders further optimization. Ac-

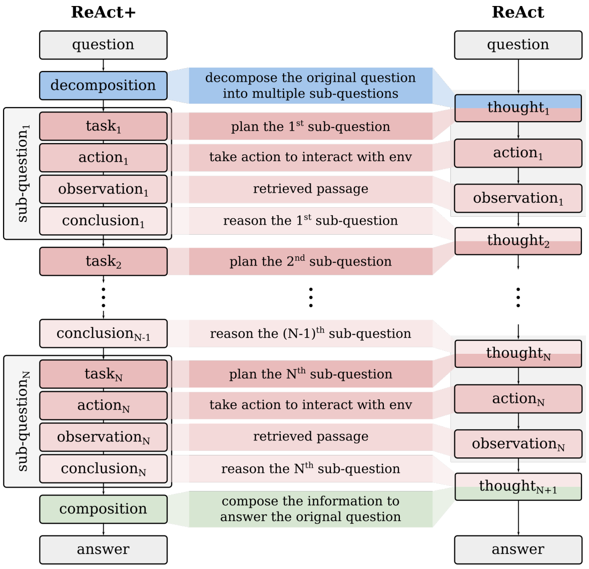

ReAct+
ReAct
question
decomposition
decompose the original question into multiple sub-questions
thought$_1$
sub-question$_1$
task$_1$
plan the 1st sub-question
action$_1$
take action to interact with env
observation$_1$
retrieved passage
conclusion$_1$
reason the 1st sub-question
thought$_2$
task$_2$
plan the 2nd sub-question
conclusion$_{N-1}$
reason the (N-1)th sub-question
thought$_N$
sub-question$_N$
task$_N$
plan the Nth sub-question
action$_N$
take action to interact with env
observation$_N$
retrieved passage
conclusion$_N$
reason the Nth sub-question
thought$_{N+1}$
composition
compose the information to answer the orignal question
answer

Figure 1: ReAct+: An equivalent variant of ReAct for open domain multi-hop question answering, where the sub-questions are explicitly decoupled.

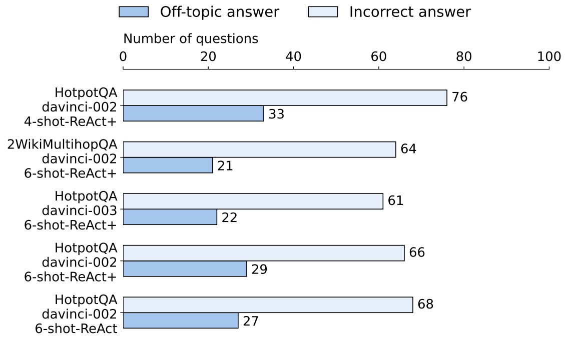


Figure 2: Statistics on off-topic and incorrect answers in ODMHQA with LLMs. We evaluated 100 randomly sampled cases for each combination of different datasets, LLMs, prompts, and methods.

cordingly, we develop ReAct+ (shown in Figure 1), an equivalent variant of ReAct better suited for multi-hop reasoning chains in ODMHQA. ReAct+ decomposes a complex question into a set of clear-cut Sub-Questions, inspired by Press et al. (2022). In ReAct+, the Sub-Question at step i is denoted by $S_i \equiv (t_i, a_i, o_i, c_i)$, where t_i signifies the planned **task**, a_i represents the selected **action**, o_i represents the **observation** from the environment, and c_i symbolizes the intermediate **conclusion**. As Figure 1 illustrates, ReAct's τ_i corresponds to ReAct+'s (c_{i-1}, t_i) for intermediate Sub-Questions. Regarding the first and last Sub-Questions, τ_1 corresponds to (D, t_1) while τ_{N+1} corresponds to (c_N, C), where D denotes **Decomposition** and C represents **Composition**. By implementing these modifications, we observe that ReAct+ marginally outperforms ReAct on ODMHQA (see Section 5.1), indicating that decomposing a complex question into Sub-Questions does not impair performance, aligning with the findings of Mishra et al. (2022).

2.2. Prevalence of Off-Topic Answers

We observe that LLMs encounter the problem of generating **off-topic answers** when attempting to solve ODMHQA. To elaborate, off-topic answers refer to cases where the generated answers are not relevant to the original questions, which is similar to the concepts of off-topic responses in dialogue systems (Malinin et al., 2016) and off-topic essays in high-stakes tests (Louis and Higgins, 2010). Notably, off-topic answers are necessarily incorrect answers. We show two cases of off-topic answers in the 3rd and 4th examples in Table 1. In the 3rd example, on-topic answers should be either "The Secret Life of Pets 2" or "Love Me Deadly", as indicated in the original question. However, the generated answer "Lover 3" does not match either of these expected on-topic answers. In the 4th example, an on-topic answer should specify the language(s), while the generated answer "Edinburgh" is a city name instead of specifying a language.

To determine the significance of the off-topic issue, we randomly sample 500 cases from the HotpotQA (Yang et al., 2018) and 2WikiMultiHopQA (Ho et al., 2020) datasets and manually label whether the LLM-generated answers are off-topic. As shown in Figure 2, off-topic answers are a prevalent issue observed across different datasets, LLMs, prompts, and methods. We notice that **approximately 1/3 of the incorrect answers are identified as off-topic answers**, which directly impairs the user experience of employing LLMs for solving ODMHQA.

2.3. Cause Analysis on Off-Topic Answers

To further investigate the issue of off-topic answers, we analyze 112 off-topic answers out of 500 cases randomly sampled from the HotpotQA dataset (Yang et al., 2018) using ReAct+. We thoroughly review the full solving history and then locate and classify the causes behind these off-topic answers. For these off-topic cases, the distribution of causes identified through this analysis is shown in Figure 3. Here are the key findings from our analysis:

- The Decomposition step accounts for 31% of the off-topic answers, where LLMs misunderstand the original question or lose the keyword during the process of decomposing the question.
- The Sub-Question steps contribute to the largest portion at 62% of the off-topic answers. We further classify these as planning errors, passage

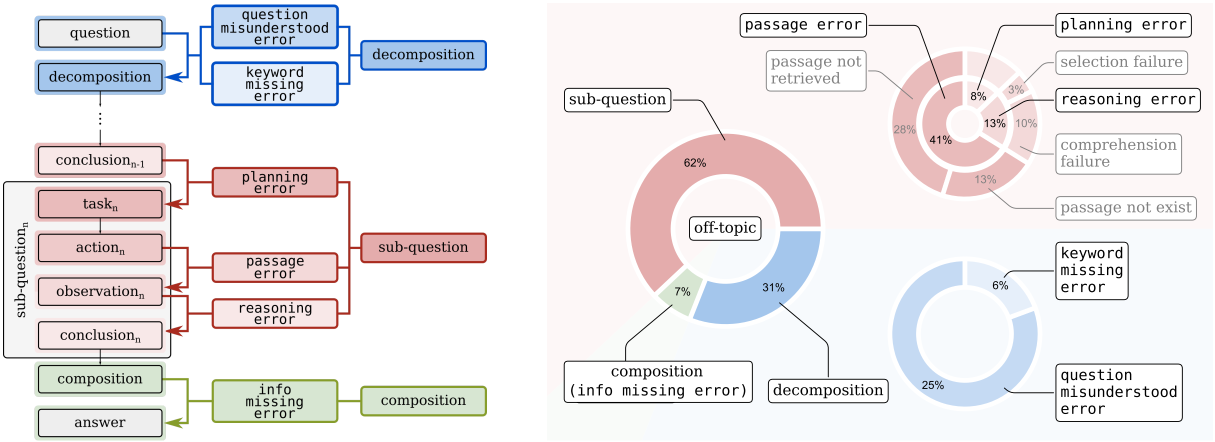


Figure 3: Cause analysis on off-topic answers. We locate and classify the causes behind 112 off-topic answers out of 500 cases randomly sampled from the HotpotQA dataset using ReAct+.

errors, and reasoning errors. Notably, passage errors (Yao et al., 2022) cover 41% of the cases, occurring when the relevant passage was not successfully retrieved or does not exist in the external database. Reasoning errors stem from two main sources: comprehension failures, where the model fails to generate the correct intermediate answer by faithfully following the reasoning chain, known as unfaithful reasoning (Lyu et al., 2023); and selection failures, where the model mistakenly chooses another answer candidate instead of the required one.

- The `Composition` step accounts for a relatively small portion at 7% of the off-topic answers. In these cases, we observe that the LLM becomes trapped in hallucination (Zhang et al., 2023), missing the key information from the original question or overlooking key evidence from intermediate conclusions, ultimately causing off-topic answers.

3. Method

In this section, we present our Dr3 mechanism (demonstrated in Figure 4), which aims to address the issue of off-topic answers, based on the analysis presented in Section 2.3. The Dr3 mechanism consists of two key modules: the Discriminator and the Corrector. The Discriminator judges whether the generated answer is on-topic by leveraging the intrinsic capabilities of LLMs. The Corrector applies heuristic rules to correct the solving history along the reversed reasoning chain (Re-Compose→Re-Solve→Re-Decompose) until the Discriminator confirms that the answer is on-topic.

3.1. Discriminator

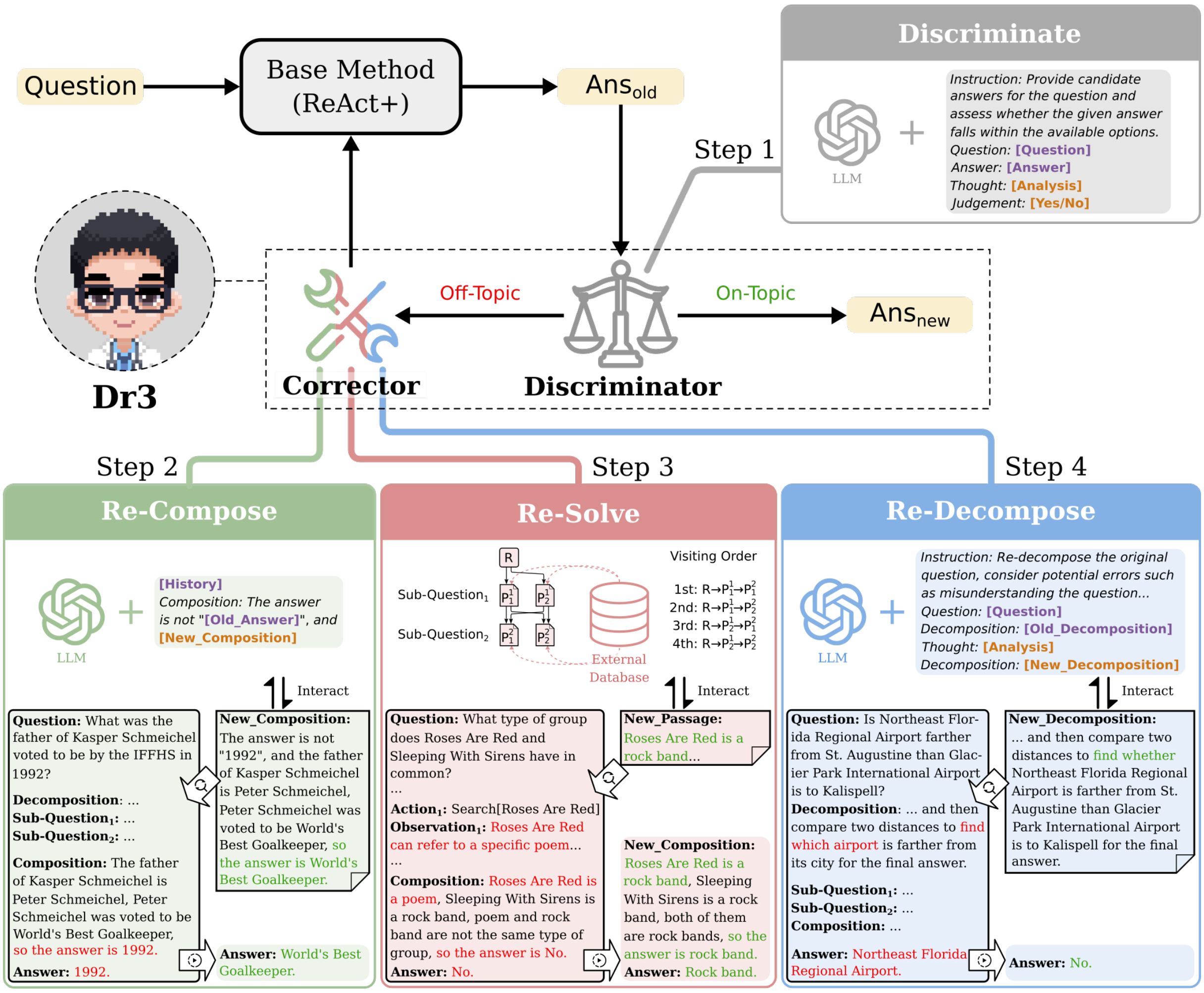
In the Discriminator, we leverage the intrinsic capabilities of LLMs to determine whether the generated answer is on-topic. This is accomplished by feeding both the original question and the LLM-generated answer into the Discriminator, which yields a binary judgment of either `YES` or `NO`, indicating whether the generated answer is on-topic or off-topic. Specifically, we design an instruction that prompts the LLM to first conceptualize candidate answers and subsequently assess whether the generated answer falls within the range of available options. If the generated answer is among the candidate answers, it should be considered on-topic; otherwise, it should be considered off-topic.

3.2. Corrector

When the generated answer, $\texttt{Ans}_{\texttt{old}}$, is identified as off-topic by the Discriminator, the Corrector revises the reasoning chain in three stages. These three stages follow the order, Re-Compose→Re-Solve→Re-Decompose, which mirrors ReAct+'s original order of reasoning, `Composition`←`Sub-Question`←`Decomposition`. At each revision stage, the Corrector generates a new answer, $\texttt{Ans}_{\texttt{new}}$, which is evaluated by the Discriminator for on-topic or off-topic. This iterative process continues either until the Discriminator approves $\texttt{Ans}_{\texttt{new}}$ as an on-topic answer, or until the entire reasoning chain has been exhausted. We introduce the Corrector's three revision stages as follows.

3.2.1. Re-Compose

The Corrector's Re-Compose stage addresses the off-topic issue that occurs in ReAct+'s original `Composition` stage. Inspired by Zheng et al. (2023), we provide a hint, h_C, to the LLM, stating "`The answer is not` [$\texttt{Ans}_{\texttt{old}}$]" at the start of the `Composition` stage. This hint encourages the LLM to reconsider the question and evidence, generating a new answer that is potentially on-topic.



Instruction: Provide candidate answers for the question and assess whether the given answer falls within the available options.
Question: **[Question]**
Answer: **[Answer]**
Thought: **[Analysis]**
Judgement: **[Yes/No]**

[History]
*Composition: The answer is not "***[Old_Answer]***", and*
[New_Composition]

Question: What was the father of Kasper Schmeichel voted to be by the IFFHS in 1992?
Decomposition: ...
Sub-Question$_1$: ...
Sub-Question$_2$: ...
Composition: The father of Kasper Schmeichel is Peter Schmeichel, Peter Schmeichel was voted to be World's Best Goalkeeper, so the answer is 1992.
Answer: 1992.

New_Composition: The answer is not "1992", and the father of Kasper Schmeichel is Peter Schmeichel, Peter Schmeichel was voted to be World's Best Goalkeeper, so the answer is World's Best Goalkeeper.

Answer: World's Best Goalkeeper.

Question: What type of group does Roses Are Red and Sleeping With Sirens have in common?
...
Action$_1$: Search[Roses Are Red]
Observation$_1$: Roses Are Red can refer to a specific poem...
...
Composition: Roses Are Red is a poem, Sleeping With Sirens is a rock band, poem and rock band are not the same type of group, so the answer is No.
Answer: No.

New_Passage: Roses Are Red is a rock band...

New_Composition: Roses Are Red is a rock band, Sleeping With Sirens is a rock band, both of them are rock bands, so the answer is rock band.
Answer: Rock band.

Instruction: Re-decompose the original question, consider potential errors such as misunderstanding the question...
Question: **[Question]**
Decomposition: **[Old_Decomposition]**
Thought: **[Analysis]**
Decomposition: **[New_Decomposition]**

Question: Is Northeast Florida Regional Airport farther from St. Augustine than Glacier Park International Airport is to Kalispell?
Decomposition: ... and then compare two distances to find which airport is farther from its city for the final answer.
Sub-Question$_1$: ...
Sub-Question$_2$: ...
Composition: ...
Answer: Northeast Florida Regional Airport.

New_Decomposition: ... and then compare two distances to find whether Northeast Florida Regional Airport is farther from St. Augustine than Glacier Park International Airport is to Kalispell for the final answer.

Answer: No.

Figure 4: Dr3 mechanism: Discriminate→Re-Compose→Re-Solve→Re-Decompose.

3.2.2. Re-Solve

The Corrector's Re-solve stage addresses the off-topic issue that occurs during ReAct+'s original Sub-Question stage. The focus here is on fixing passage errors, which was identified as the primary cause for off-topic answers in the analysis conducted in Section 2.3. For Sub-Question S_i, we first replace the passage in the observation, o_i, based on retrieval probabilities from the IR system. Subsequently, we utilize ReAct+ to resolve S_i and obtain a new answer, Ans_{new}. This procedure repeats for S_i until the Discriminator approves that Ans_{new} is an on-topic answer. If the number of replacements reaches the predefined threshold, T_D, without generating an on-topic answer, we iterate this procedure backwards for Sub-Question, S_{i-1}. This continues backwards until an on-topic answer is obtained or the first Sub-Question, S_1, is addressed.

3.2.3. Re-Decompose

The Corrector's Re-Decompose stage addresses the off-topic issue that occurs in ReAct+'s original Decomposition stage. Inspired by Xi et al. (2023), we leverage the intrinsic capabilities of LLMs to refine the Decomposition. To achieve this, we input the original question along with the existing Decomposition into the LLM. Our prompts guide the LLM to revise the Decomposition in cases of errors including misunderstanding the question and missing keywords. Subsequently, the LLM generates a new Decomposition that replaces the old one, enabling ReAct+ to continue execution and generate a new answer, Ans_{new}.

4. Experiments

4.1. Datasets and Metrics

We present the results obtained from two popular datasets: HotpotQA and 2WikiMultiHopQA. **HotpotQA** (Yang et al., 2018) is a 2-hop QA dataset built from Wikipedia passages where reasoning chains are formed between passage pairs. We use the same data following Yao et al. (2022) as our test dataset, which randomly samples 500 cases from Dev split. For each case, we only provide the question without the paired passages according to the open domain setting. **2WikiMultiHopQA** (Ho et al., 2020) constructs multi-hop QA cases by combining Wikipedia articles and Wikidata knowledge. Compared to HotpotQA, 2WikiMultiHopQA is more challenging because it includes four question types: comparison, inference, compositional, and bridge-comparison. Similar to Yao et al. (2022) and Khattab et al. (2022), we randomly sample 500 question-answer pairs from the Dev split as the testing dataset. Following previous work in QA (Yang et al., 2018; Xu et al., 2023), we evaluate using the metrics of token F1 score, exact match (EM), and cover exact match (Cover EM).

4.2. Baselines

The baseline methods used for comparison are briefly described as follows.

- **Zero-Shot** (Kojima et al., 2022): The zero-shot approach requires the LLM to output the answer to the question without any relevant example.
- **Few-Shot** (Brown et al., 2020): The Few-Shot approach prompts the LLM with a few relevant question–answer examples, demonstrating the procedure for solving this type of task.
- **CoT** (Wei et al., 2022): The Chain-of-Thought (CoT) approach facilitates the LLM to generate coherent intermediate reasoning steps that mimic a step-by-step thought process by prompting "`Let's think step-by-step`".
- **ReAct** (Yao et al., 2022): The Reasoning-Acting (ReAct) approach enhances reasoning-only LLMs by incorporating acting capabilities through the use of external tools, which defines the reasoning pattern as a sequence of interleaved Thought-Action-Observation steps (see Section 2.1).
- **ReAct+** (Ours): The ReAct+ approach modifies the reasoning pattern of ReAct to fit multi-hop question answering, which involves interleaved Task-Action-Observation-Conclusion steps (see Section 2.1).

4.3. Implementation Details

For the Re-Solve step, we set the maximum number of replaced passages T_D to 3 for each `Sub-Question`. We use the off-the-shelf ColBERTv2 (Santhanam et al., 2022) following Khattab et al. (2022) as the IR system, which encodes the Wikipedia corpus by passage. For the Re-Decompose step, we use 6 question-answer examples in prompts. Within each step, we set the maximum number of continuous `Sub-Question` to 7, following Yao et al. (2022). All methods are implemented using *text-davinci-002* as the LLM, in line with Yao et al. (2022) in September 2023. The detailed prompts can be found in Appendix A, B, and C.

5. Results and Analysis

In this section, we first present the overall results of our Dr3 mechanism. Following that, we evaluate the performance of the Discriminator in detecting off-topic answers. Next, we assess the Corrector on lowering the off-topic ratio. Additionally, we analyze the relationship between the off-topic issue and factors including the number of `Sub-Questions`, and multi-hop question types.

5.1. Main Results

	HotpotQA			2WikiMultiHopQA		
Method	**EM↑**	**Cover EM↑**	**F1↑**	**EM↑**	**Cover EM↑**	**F1↑**
Zero-Shot	8.80	27.80	22.65	4.60	29.8	17.08
Few-Shot	21.20	28.00	34.41	20.40	22.80	25.45
CoT	30.40	37.60	43.93	28.80	33.60	36.37
ReAct	30.80†	-	-	34.40	41.80	43.24
ReAct+ (Ours)	31.00	36.60	42.21	35.60	43.60	45.39
Dr3 (Ours)	**33.80**	**40.00**	**46.53**	**38.80**	**46.60**	**48.36**

Table 2: Main results on the HotpotQA and 2WikiMultiHopQA datasets. The result with † is borrowed from Yao et al. (2022).

In this subsection, we evaluate the overall performance of our Dr3 mechanism on the HotpotQA and 2WikiMultiHopQA datasets, compared to five baseline methods described in Section 4.2. As shown in Table 2, our Dr3 approach outperforms existing methods across evaluation metrics of EM, Cover EM, and F1 score. Specifically, compared to the best baseline ReAct+, our Dr3 approach exhibits significant improvements: on HotpotQA, we observe absolute enhancements of 2.80% in EM, 3.40% in Cover EM, and 4.32% in F1 score. Meanwhile, on 2WikiMultiHopQA, we achieve even larger gains of 3.20% in EM, 3.00% in Cover EM, and 2.97% in F1 score. These compelling results clearly demonstrate the effectiveness of our Dr3 mechanism for improve the performance on ODMHQA.

5.2. Performance of Discriminator

In this subsection, we examine the performance of the Discriminator in detecting off-topic answers. **Discriminator is comparable to human judgment in detecting off-topic answers.** As shown in Table 3, the Discriminator achieves an accuracy of 92.77% on the HotpotQA dataset, which manifests its remarkable capability in identifying

Accuracy	Precision	Recall	F1
92.77	83.76	85.21	84.82

Table 3: Performance of Discriminator in detecting off-topic answers. Human evaluation as the ground truth on the HotpotQA dataset, while precision, recall, F1 are for the off-topic class.

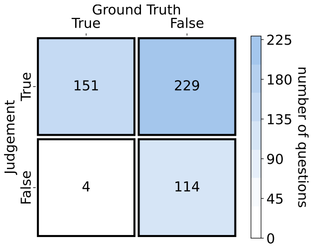


Figure 5: Confusion matrix of QA correctness by Discriminator. Evaluation performed on the HotpotQA dataset, while "True" means correct and "False" means incorrect.

off-topic answers, closely resembling human judgment. Moreover, the high F1 score of 84.82% further emphasizes the Discriminator's capability to recall a majority of the off-topic answers while maintaining a high precision.

Discriminator identifies 1/3 off-topic answers and rarely misjudges correct answers. Figure 5 illustrates the confusion matrix of the Discriminator in judging QA correctness on the HotpotQA dataset. The Discriminator has a sensitivity of 97.4%($\frac{151}{151+4}$), indicating that the Discriminator rarely misjudges correct answers as incorrect. Meanwhile, it has a specificity of 33.2%($\frac{114}{114+229}$), signifying that it successfully identifies a substantial portion of incorrect answers. Therefore, the benefit of identifying incorrect answers outweighs the misjudgments.

5.3. Performance of Corrector

In this subsection, we conduct ablation studies on three individual Corrector modules to investigate their effectiveness in improving question answering and alleviating the off-topic issue. The results are shown in Table 4. **Re-Solve is the most effective module, exhibiting a 1.4% EM improvement.** This significant EM improvement can be attributed to Re-Solve identifying passage errors as the primary cause of off-topic answers (see Section 2.3). **Re-Compose (+1.0% EM) appears to be more effective than Re-Decompose (+0.4% EM).** This is unexpected that Re-Compose is so effective given `Composition` errors comprise the smallest ratio (7%) as shown in Section 2.3. We conjecture this occurs because Re-Compose not only fixes `Composition` errors but also encourages the LLM to re-examine the full solving history. By thoroughly reassessing the question and evidence, Re-Compose can implicitly fix reading comprehension and `Decomposition` errors as well. To demonstrate this phenomenon, we present 2 cases in Appendix D.

5.4. The Effect of Sub-Question Numbers

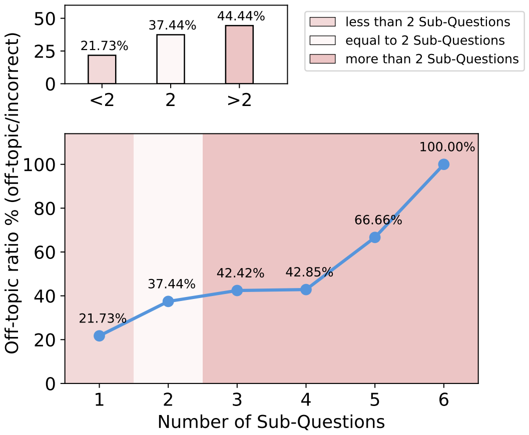


Figure 6: Off-topic ratio versus number of `Sub-Questions`. Evaluation conducted on 108 off-topic answers from the HotpotQA dataset.

In this subsection, we discuss the relationship between off-topic answers and the number of tasks in `Sub-Questions`. As the line chart in Figure 6 illustrates, the off-topic ratio increases monotonically as the number of tasks rises. Ideally, the reasoning chain should contain exactly 2 `Sub-Questions`, since the questions from the HotpotQA dataset are designed to be 2-hop. Accordingly, depending on whether the number of tasks is less than, equal to, or greater than 2, we organize the same data into three groups in the bar chart of Figure 6. For the 1-hop reasoning chains, in which off-topic answers account for 21.73% of cases, we notice that ReAct+ tends to conclude the answer prematurely if it is on-topic. For reasoning chains with more than 2 hops, where off-topic answers occur 44.44% of the time, we observe that ReAct+ becomes trapped in self-correction cycles. This unavoidably adds noise to the context, causing the original question to be forgotten and resulting in off-topic answers.

5.5. The Effect of Question Types

In this subsection, we examine the relationship between off-topic answers and the four question types on the 2WikiMultiHopQA dataset. The results are shown in Figure 7. The Comparison type

Re-Decompose	Re-Solve	Re-Compose	EM↑	F1↑	Off-Topic Ratio (Discriminator)↓	Off-Topic Ratio (Expert)↓
✗	✗	✗	31.00	42.21	23.69	23.09
✗	✗	✔	32.00	43.11	18.07	-
✗	✔	✔	33.40	45.51	9.83	-
✔	✔	✔	**33.80**	**46.53**	**7.42**	**10.24**

Table 4: Ablation studies of the Corrector modules. Evaluation performed on the HotpotQA dataset.

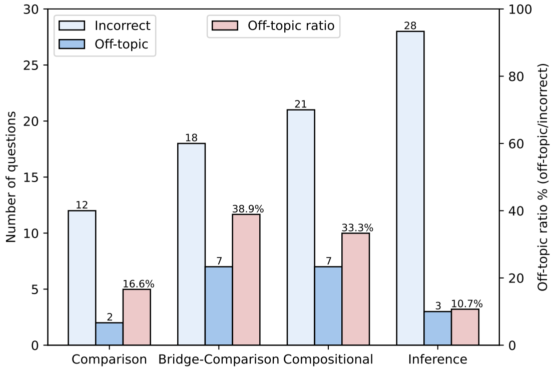


Figure 7: Off-topic answers versus question types. We randomly sample 30 questions per type on the 2WikiMultiHopQA dataset.

has the second lowest off-topic ratio, as the answer is already provided within the original question. For example, in a question like "Who is older, A or B?", the answer can only be either A or B, both options being clearly stated in the question itself. The Bridge-Comparison type has the highest off-topic ratio. For instance, in a question like "Which film has the director born first, A or B?", ReAct+ is prone to answering the name of the director rather than the name of the film, even though the correct answer must be either A or B. We hypothesize that the Bridge-Comparison type requires a minimum of 4 tasks to be properly solved, as extended reasoning causes the original question to be forgotten. The Inference type exhibits the lowest off-topic ratio but the highest error ratio. We observe that ReAct+ tends to prematurely terminate the reasoning chain when an intermediate answer is on-topic. For example, in a question like "Who is A's grandfather?", where the first retrieved passage contains A's father B, the corresponding conclusion might erroneously regard B as A's grandfather.

6. Related Work

Large Language Models on Open Domain Multi-Hop Question Answering. LLMs have shown outstanding capabilities in reasoning, planning, and utilizing tools, all of which are indispensable for performing natural language question answering (He et al., 2022; Trivedi et al., 2022). The pioneering work CoT (Wei et al., 2022) proposed to generate intermediate reasoning steps to answer questions that require complex reasoning capabilities. Inspired by CoT, Self-Consistency (Wang et al., 2023c) and Complexity-Based Prompting (Fu et al., 2022) sampled multiple reasoning trajectories to vote for the final answer. Despite their performance enhancement, these voting-based CoT methods are resource-intensive and time-consuming. Meanwhile, Press et al. (2022) pointed out that answering a compositional question as a whole is often more difficult than correctly answering its individual subquestions. Based on this insight, Self-Ask (Press et al., 2022), PS (Wang et al., 2023b), ReAct (Yao et al., 2022), and DSP (Khattab et al., 2022) decomposed the compositional question into a series of sub-questions and employed external knowledge to prevent factual errors when answering individual sub-questions. Our Dr3 is constructed based on the ReAct framework, as it was the most influential among the aforementioned works and also inspired subsequent LLM-Agent research (Mialon et al., 2023; Wang et al., 2023a).

Post-Hoc Correction. Post-hoc correction refers to the process of refining the output from LLMs after it has been generated, without making any modification to the model parameters (Pan et al., 2023). Self-Refine (Madaan et al., 2023) iteratively polished outputs by incorporating LLM feedback for dialogue and code generation tasks. Auto-Post-Editing (Raunak et al., 2023) and RCI (Kim et al., 2023) demonstrated similar ideas of using LLMs to provide advice then using that advice to prompt higher quality outputs, which were applied to translation and computer tasks. DIN-SQL (Pourreza and Rafiei, 2023) proposed generic and gentle correction modules to fix bugs and potential issues respectively in the text-to-SQL task. When it comes to post-hoc correction methods for QA tasks, Verify-and-Edit (Zhao et al., 2023) and LLM-AUGMENTER (Peng et al., 2023) employed external knowledge to enhance the quality of the generated text. In open-domain QA, SearChain (Xu et al., 2023) utilized a trained small model to improve the quality of the retrieved passage.

While previous research concentrated on correcting factual errors in the reasoning process, our Dr3 focuses on alleviate the off-topic issue. Furthermore, our Dr3 does not rely on any voting-based mechanism or additional tools for fact-checking sub-questions.

7. Conclusion

In this paper, we have identified the crucial issue of off-topic answers that occurs when utilizing Large Language Models (LLMs) to tackle open domain multi-hop question answering (ODMHQA). Our proposed solution, the Discriminate→Re-Compose→Re-Solve→Re-Decompose (Dr3) mechanism, effectively harnesses the intrinsic capabilities of LLMs to detect and correct off-topic answers by performing step-wise revisions along the reversed reasoning chain. Through comprehensive experiments conducted on the HotpotQA and 2WikiMultiHopQA datasets, we have demonstrated the effectiveness of our approach in alleviating the off-topic issue. We anticipate that our work not only provides a practical solution to the issue of off-topic answers but also serves as a catalyst for future research in this area.

8. Acknowledgments

We would like to thank the anonymous reviewers for their valuable feedback. This work was supported by the NSFC (No. 62376120, 61936012, 62206126).

9. Bibliographical References

Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, et al. 2023. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. *arXiv preprint arXiv:2302.04023*.

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.

Yair Feldman and Ran El-Yaniv. 2019. Multi-hop paragraph retrieval for open-domain question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2296–2309.

Yao Fu, Hao Peng, Ashish Sabharwal, Peter Clark, and Tushar Khot. 2022. Complexity-based prompting for multi-step reasoning. In *The 11th International Conference on Learning Representations*.

Shibo Hao, Tianyang Liu, Zhen Wang, and Zhiting Hu. 2023. Toolkengpt: Augmenting frozen language models with massive tools via tool embeddings. *arXiv preprint arXiv:2305.11554*.

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. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6609–6625.

Cheng-Yu Hsieh, Si-An Chen, Chun-Liang Li, Yasuhisa Fujii, Alexander Ratner, Chen-Yu Lee, Ranjay Krishna, and Tomas Pfister. 2023. Tool documentation enables zero-shot tool-usage with large language models. *arXiv preprint arXiv:2308.00675*.

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*.

Geunwoo Kim, Pierre Baldi, and Stephen McAleer. 2023. Language models can solve computer tasks. *arXiv preprint arXiv:2303.17491*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in Neural Information Processing Systems*, 35:22199–22213.

Annie Louis and Derrick Higgins. 2010. Off-topic essay detection using short prompt texts. In *Proceedings of the NAACL HLT 2010 5th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 92–95.

Qing Lyu, Shreya Havaldar, Adam Stein, Li Zhang, Delip Rao, Eric Wong, Marianna Apidianaki, and Chris Callison-Burch. 2023. Faithful chain-of-thought reasoning. *arXiv preprint arXiv:2301.13379*.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri

Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. 2023. Self-refine: Iterative refinement with self-feedback. *arXiv preprint arXiv:2303.17651*.

Andrey Malinin, Rogier Van Dalen, Kate Knill, Yu Wang, and Mark Gales. 2016. Off-topic response detection for spontaneous spoken english assessment. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 1075–1084.

Vaibhav Mavi, Anubhav Jangra, and Adam Jatowt. 2022. A survey on multi-hop question answering and generation. *arXiv preprint arXiv:2204.09140*.

Grégoire Mialon, Roberto Dessì, Maria Lomeli, Christoforos Nalmpantis, Ram Pasunuru, Roberta Raileanu, Baptiste Rozière, Timo Schick, Jane Dwivedi-Yu, Asli Celikyilmaz, et al. 2023. Augmented language models: a survey. *arXiv preprint arXiv:2302.07842*.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, Yejin Choi, and Hannaneh Hajishirzi. 2022. Reframing instructional prompts to gptk's language. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, pages 589–612.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. 2021. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*.

Liangming Pan, Michael Saxon, Wenda Xu, Deepak Nathani, Xinyi Wang, and William Yang Wang. 2023. Automatically correcting large language models: Surveying the landscape of diverse self-correction strategies. *arXiv preprint arXiv:2308.03188*.

Baolin Peng, Michel Galley, Pengcheng He, Hao Cheng, Yujia Xie, Yu Hu, Qiuyuan Huang, Lars Liden, Zhou Yu, Weizhu Chen, et al. 2023. Check your facts and try again: Improving large language models with external knowledge and automated feedback. *arXiv preprint arXiv:2302.12813*.

Mohammadreza Pourreza and Davood Rafiei. 2023. Din-sql: Decomposed in-context learning of text-to-sql with self-correction. *arXiv preprint arXiv:2304.11015*.

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 preprint arXiv:2210.03350*.

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.

Vikas Raunak, Amr Sharaf, Hany Hassan Awadallah, and Arul Menezes. 2023. Leveraging gpt-4 for automatic translation post-editing. *arXiv preprint arXiv:2305.14878*.

Jingqing Ruan, Yihong Chen, Bin Zhang, Zhiwei Xu, Tianpeng Bao, Guoqing Du, Shiwei Shi, Hangyu Mao, Xingyu Zeng, and Rui Zhao. 2023. Tptu: Task planning and tool usage of large language model-based ai agents. *arXiv preprint arXiv:2308.03427*.

Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. 2022. Colbertv2: Effective and efficient retrieval via lightweight late interaction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3715–3734.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, et al. 2023a. A survey on large language model based autonomous agents. *arXiv preprint arXiv:2308.11432*.

Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. 2023b. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023c. Self-consistency improves chain of thought reasoning in language models. In *The 11th International Conference on Learning Representations*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Zhiheng Xi, Senjie Jin, Yuhao Zhou, Rui Zheng, Songyang Gao, Tao Gui, Qi Zhang, and Xuanjing Huang. 2023. Self-polish: Enhance reasoning in large language models via problem refinement. *arXiv preprint arXiv:2305.14497*.

Shicheng Xu, Liang Pang, Huawei Shen, Xueqi Cheng, and Tat-seng Chua. 2023. Search-in-the-chain: Towards the accurate, credible and traceable content generation for complex knowledge-intensive tasks. *arXiv preprint arXiv:2304.14732*.

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 11th International Conference on Learning Representations*.

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, Longyue Wang, Anh Tuan Luu, Wei Bi, Freda Shi, and Shuming Shi. 2023. Siren's song in the ai ocean: A survey on hallucination in large language models. *arXiv preprint arXiv:2309.01219*.

Ruochen Zhao, Xingxuan Li, Shafiq Joty, Chengwei Qin, and Lidong Bing. 2023. Verify-and-edit: A knowledge-enhanced chain-of-thought framework. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, volume 1.

Chuanyang Zheng, Zhengying Liu, Enze Xie, Zhenguo Li, and Yu Li. 2023. Progressive-hint prompting improves reasoning in large language models. *arXiv preprint arXiv:2304.09797*.

Fengbin Zhu, Wenqiang Lei, Chao Wang, Jianming Zheng, Soujanya Poria, and Tat-Seng Chua. 2021. Retrieving and reading: A comprehensive survey on open-domain question answering. *arXiv preprint arXiv:2101.00774*.

A. ReAct+ Prompt

# Instruction

You are a question-answering agent. To answer a difficult Question, you need to perform **Decomposition** to divide it into several tasks, solve them and integrate the information for the answer.

To solve each task, you need to use interleaving **Task**, **Action**, **Observation**, and **Conclusion** steps. The steps are:

1. **Task**: a sub-problem to be solved from Decomposition and the previous Conclusion.
2. **Action**: Search[Query] to retrieve a document corresponding to the Query.
3. **Observation**: the retrieved document by the Action.
4. **Conclusion**: the Task result according to the Observation.

According to the **Decomposition**, when all the necessary tasks are finished, you need to execute **Composition** and then answer the Question with **Finish[Answer]**. The steps are:

1. **Composition**: the composition of the information from all the tasks.
2. **Finish[Answer]**: the final Answer to the Question.

# We demonstrate a case here; all cases can be found in our codes.

Question: Musician and satirist Allie Goertz wrote a song about the “The Simpsons” character Milhouse, who Matt Groening named after who?

Decomposition: The question simplifies to “The Simpsons” character Milhouse is named after who. I only need to search Milhouse and find who it is named after.

# Sub-Question 1

Task 1: I need to search Milhouse and find who it is named after.

Action 1: Search[Milhouse]

Observation 1: Milhouse Mussolini Van Houten is a recurring character in the Fox animated television series The Simpsons voiced by Pamela Hayden and created by Matt Groening.

Conclusion 1: The paragraph does not tell who Milhouse is named after.

# Sub-Question 2

Task 2: I can search Milhouse named after whom instead to find who it is named after.

Action 2: Search[Milhouse named after whom]

Observation 2: Milhouse was named after U.S. president Richard Nixon, whose middle name was Milhous.

Conclusion 2: Milhouse was named after U.S. president Richard Nixon.

Composition: Milhouse was named after U.S. president Richard Nixon, so the answer is Richard Nixon.

Finish: [Richard Nixon]

B. Discriminator Prompt

# Instruction

You will be given QUESTION and ANSWER. You need to identify the possible answer to the QUESTION, and then check whether the ANSWER is the possible ANSWER.

For every case, you must think firstly in **THOUGHT**, and then output your **JUDGMENT**, with the format:

THOUGHT: (your analysis here).

JUDGMENT: YES / NO

# We demonstrate a case here; all cases can be found in our codes.

QUESTION: When was the Man Falls in Love born on?

ANSWER: July 5, 1984

THOUGHT: The answer to the QUESTION can be a year, date or time range. The ANSWER “July 5, 1984” is a date. So the answer is YES.

JUDGMENT: YES

C. Re-Decompose Prompt

# Instruction

You are a teacher, and your student is in the process of decomposing a multi-hop QUESTION into several sub-questions, solving them to arrive at the final answer.

However, there are some mistakes in the **STUDENT DECOMPOSITION** that the student cannot answer the question. e.g., **misunderstanding the question**, **the divided ones are too ambiguous or difficult to lead to the final answer**, and so on.

For every case, you must analyze the question first in the **ANALYSIS**, which contains the understanding of the question, and the weaknesses of the student decomposition. After analysis, give a targeted suggestion based on the understanding of the problem and students' weaknesses in the **ADVICE**. Finally, refer to the advice, and give a more specific, correct and fluent decomposition in the **NEW DECOMPOSITION.**

The Input would be in the following format:

QUESTION: (the difficult question)

STUDENT DECOMPOSITION: (the decomposition from the student)

The Output must be in the following format:

ANALYSIS: (analysis here. Analyze the question and student's decomposition)

ADVICE: (advice here. Refer to the analysis, give targeted advice)

NEW DECOMPOSITION: (new decomposition here. Refer to the advice, give a more specific, correct and fluent version than the STUDENT DECOMPOSITION)

# We demonstrate a case here; all cases can be found in our codes.

QUESTION: Is Level 3 Communications headquartered further west than Advanced Micro Devices?

STUDENT DECOMPOSITION: I need to find the headquarters location of Advanced Micro Devices and Level 3 Communications, and then integrate the information to identify which company is headquartered further west for the final answer

ANALYSIS: The answer to the question should be either yes or no, if Level 3 Communications is headquartered further west than Advanced Micro Devices, the answer is yes, other is no. The student decomposition tries to ”identify which company”, the answer delivered from the decomposition will be a company, neither yes nor no, which causes the error.

ADVICE: I think the last step should find whether Level 3 Communications is headquartered further west, instead of ”identify which company...”.

NEW DECOMPOSITION: I need to find the headquarters location of Advanced Micro Devices and Level 3 Communications, and then find whether Level 3 Communications is headquartered further west for the final answer.

D. Case Study

In Table 5, we demonstrate two cases in which off-topic answers are corrected by our Dr3 mechanism. Specifically, these two cases differ in that the error in the upper case occurs in the `Sub-Question` stage, while the error in the lower case occurs in the `Decomposition` stage. However, both can be corrected by the same Re-Compose strategy of our Dr3 mechanism.

Step	ReAct+	Dr3
Question	What nationality is the man who developed the dictator game?	
Decomposition	I need to find the nationality of the man who developed the dictator game, which is the final answer.	
Sub-Question 1	- Find the man who developed the dictator game. - Answer: Michael Pagano # Correct	
Sub-Question 2	- Find the nationality of Michael Pagano. - Answer: From Los Angeles, California # "Los Angeles, California" is a place, not a nationality.	
Composition	The man who developed the dictator game is Michael Pagano, Michael Pagano is from Los Angeles, California, so the answer is Los Angeles, California.	No, the answer is not "Los Angeles, California", and the answer should be the nationality, so the answer is American.
Answer	Los Angeles, California	American
Question	Philip Savage served as Director of Player Personnel for the Baltimore Ravens under what general manager who was inducted into both the College and Pro Football Halls of Fame?	
Decomposition	I need to find the general manager of Baltimore Ravens during Philip Savage served as Director of Player Personnel, and then find whether the general manager was inducted into both the College and Pro Football Halls of Fame for the final answer. # The answer should be the person who satisfies the two requirements.	
Sub-Question 1	- Find the general manager of Baltimore Ravens during he served as Director of Player Personnel. - Answer: Ozzie Newsome # Correct	
Sub-Question 2	- Find whether he was inducted into both the College and Pro Football Halls of Fame - Answer: Yes # Correct	
Composition	The general manager is Ozzie Newsome, who was inducted into both the College and Pro Football Halls of Fame, so the answer is Yes.	No, the answer is not "Yes", and Ozzie Newsome is the manager and was inducted into both the College and Pro Football Halls of Fame, so the answer is Ozzie Newsome.
Answer	Yes	Ozzie Newsome

Table 5: Two cases where the error is fixed in the Re-Compose stage. The error in the upper case occurs in the `Sub-Question` stage, while the error in the lower case occurs in the `Decomposition` stage.