DRS: Deep Question Reformulation With Structured Output

Anonymous ACL submission

Abstract

Question answering represents a core capability of large language models (LLMs). However, when individuals encounter unfamiliar knowledge in texts, they often formulate questions that the text itself cannot answer due to insufficient understanding of the underlying information. Recent studies reveal that while LLMs can detect unanswerable questions, they struggle to assist users in reformulating these questions. Even advanced models like GPT-3.5 demonstrate limited effectiveness in this regard. To address this limitation, we propose DRS: **D**eep Question **R**eformulation with **S**tructured Output, a novel zero-shot method aimed at enhancing LLMs' ability to assist users in reformulating questions to extract relevant information from new documents. DRS combines the strengths of LLMs with a DFS-based algorithm to iteratively explore potential entity combinations and constrain outputs using predefined entities. This structured approach significantly enhances the reformulation capabilities of LLMs. Comprehensive experimental evaluations demonstrate that DRS improves the reformulation accuracy of GPT-3.5 from 23.03% to 70.42%, while also enhancing the performance of open-source models, such as GEMMA2-9B, from 26.35% to 56.75%.

1 Introduction

Question answering has emerged as a fundamental capability of large language models (LLMs), with recent advances from GPT-3 and InstructGPT to GPT-4 (Brown et al., 2020; Ouyang et al., 2022; OpenAI et al., 2024) demonstrating remarkable improvements on various benchmarks (Tan et al., 2023; Wang, 2022; Nassiri and Akhloufi, 2023). However, when humans encounter unfamiliar knowledge domains, they frequently pose questions that cannot be directly answered from the available text. Recent studies indicate that over 30% of questions in real-world scenarios fall into this category, significantly impacting information access and learning efficiency (Gao et al., 2023a; Yu et al., 2023). More importantly, industrial experiments have shown that effectively reformulating such questions can dramatically improve user experience and task completion rates in virtual assistant systems, with potential impact on millions of users (Faustini et al., 2023).

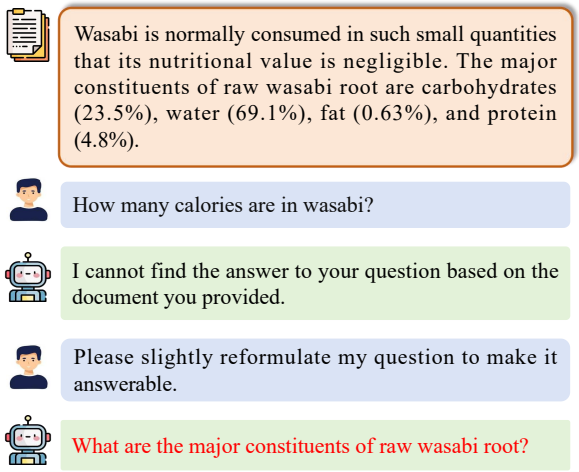

Figure 1: An example of question reformulation using large language models.

A successful question reformulation must satisfy two key criteria: (1) the reformulated question should be answerable based on the given text, and (2) it should preserve the core entities and intent of the original question, ensuring users obtain the information they actually seek. Consider the following example shown in Figure 1: when presented with a text stating "Wasabi is normally consumed in such small quantities that its nutritional value is negligible. The major constituents of raw wasabi root are carbohydrates (23.5%), water (69.1%), fat (0.63%), and protein (4.8%)", users might ask "How many calories are in wasabi?". While this question cannot be directly answered, a well-reformulated version would be "What are the major constituents of raw wasabi root?", which maintains the user's interest in wasabi's composi-

tion while being answerable from the text.

Previous approaches to handling unanswerable questions broadly fall into three categories: (1) detection methods that focus on identifying unanswerable questions (Rajpurkar et al., 2018; Asai and Choi, 2020; Sulem et al., 2021, 2022), (2) clarification methods that seek additional information from users (Deng et al., 2024; Kim et al., 2023), and (3) reformulation methods that attempt to modify questions into answerable forms (Zhao et al., 2024). While recent advances in LLMs have improved performance in question detection and clarification, question reformulation remains challenging due to the difficulty in balancing answerability and intent preservation. Even powerful models like GPT-3.5 achieve only 23.03% accuracy in reformulation tasks, highlighting the significant room for improvement in this area.

To address these challenges, we propose DRS (**D**eep Question **R**eformulation with **S**tructured Output), a zero-shot method that enables LLMs to effectively reformulate unanswerable questions. DRS addresses the key challenges through three innovations: (1) a systematic entity-driven approach that ensures intent preservation by explicitly tracking and maintaining key entities from the original question, (2) a structured output framework that ensures answerability by constraining the generation process to incorporate specific entities and generate questions aligned with the corresponding statements derived from the source document, and (3) an efficient DFS-based search strategy, coupled with a candidate question evaluation mechanism, enhances the method's effectiveness and practicality for real-world applications. Unlike previous methods that often sacrifice one aspect for another, DRS achieves strong performance on both answerability and entity preservation simultaneously.

We conduct extensive experiments on six diverse datasets, comparing DRS with multiple baseline approaches across different types of questions and domains. The results demonstrate that DRS significantly improves reformulation accuracy across all tested LLMs. Additionally, we introduce a more reliable evaluation framework using GPT-4O-MINI, replacing the previous LLAMA2-7B evaluator to ensure more accurate assessment of reformulation quality.

Our contributions in this paper are threefold:

(i) We propose DRS, a zero-shot method that enables LLMs to effectively reformulate unanswerable questions through entity-driven search and structured outputs.

(ii) We demonstrate through extensive experiments that DRS significantly outperforms existing methods, improving reformulation accuracy by over 100% across various LLMs and datasets.

(iii) We introduce an improved evaluation framework based on GPT-4O-MINI, providing more reliable assessment of question reformulation quality.

2 Methodology

In this paper, we propose a new zero-shot method called DRS: **D**eep Question **R**eformulation with **S**tructured Output, which significantly improves LLMs' ability to reformulate unanswerable questions based on a given document, helping users obtain desired answers in unfamiliar knowledge domains.

Our DRS method combines DFS (Depth-First Search) algorithm with LLMs to address challenges posed by human-generated unanswerable questions involving multiple entities. Directly inputting all entities into LLMs for question reformulation risks overlooking the lack of meaningful relationships between them, leading to unreliable or incoherent answers. To mitigate this, our DFS-based approach systematically explores semantically related entity combinations, ensuring the generation of meaningful questions. Additionally, by controlling the search depth and iteration count, our DRS method strikes a balance between computational efficiency and accuracy, yielding high-quality results with minimal overhead.

Our approach consists of three main steps: (i) Entity extraction and filtering. (ii) DFS combination search and structured question generation. (iii) Candidate question re-evaluation. The detailed process of our zero-shot DRS method is illustrated in Figure 2.

2.1 Entity Extraction and Filtering

When people encounter a text with unfamiliar knowledge, they usually raise questions about the key entities in the text, indicating their desire to understand more about these entities. Therefore, when reconstructing questions, we focus on the key entities from the original question. This ensures that the reconstructed question addresses the content people genuinely care about, rather than an arbitrary, overly simple new question.

For instance, consider the question: "When does

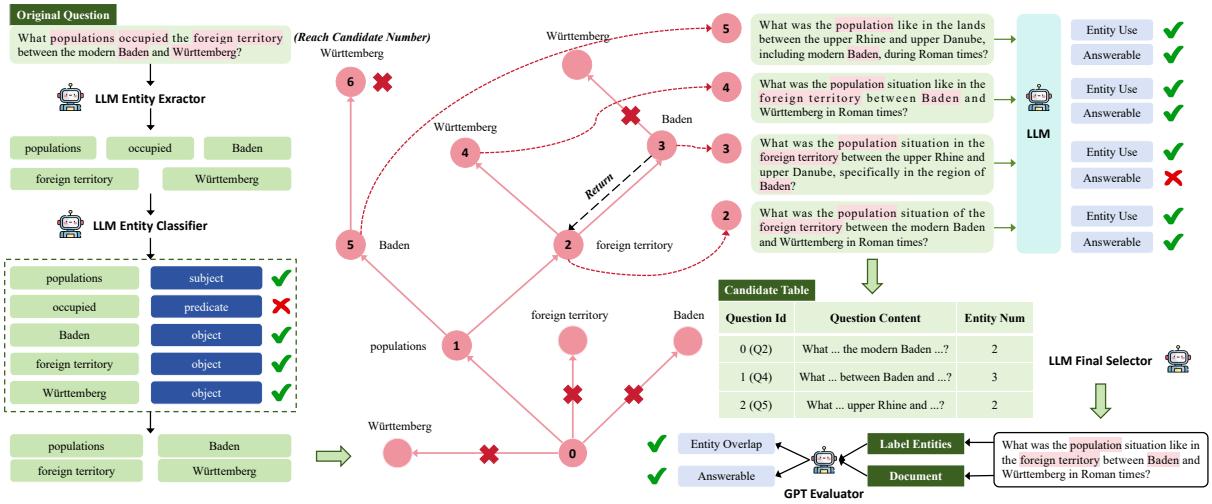

Figure 2: The complete process of our zero-shot DRS method, which mainly contains three parts - entity extraction and filtering, dfs combination search and structured question generation, and candidate question re-evaluation.

Rainer Hertrich, the German co-head of EADS, step down?". The key entities in this example are "Rainer Hertrich", "German", and "EADS", which represent the core subject and its relevant modifiers. While modern LLMs are capable of extracting important entities from a question, they sometimes include verb phrases such as "step down", which, although significant to the context, do not qualify as entities. This tendency can lead to the inclusion of extraneous elements, resulting in less precise entity identification and subsequently higher error rates in question reformulation task.

To address this limitation, we add a classification framework that categorizes entities into five semantic roles based on their function within the question. This approach facilitates the identification of core entities that align with the user's intent while filtering out less critical elements, thereby reducing ambiguities in reformulation and enhancing overall precision.

Therefore, this process involves two following sub-steps:

(i) We use a simple and direct zero-shot prompt to have the large language model extract all entities it considers important from the original question. The goal is to minimize the omission of any potentially important entities.

(ii) We then apply the large language model again as an effective entity classifier to classify the previously extracted entities into five categories: subject, object, predicate, attribute, and others. We retain all entities classified as subject, object, and attribute, as these are the most important components of the question, and discard the others.

2.2 DFS Combination Search and Structured Question Generation

After obtaining reliable entities, we proceed with question reformulation. We use the DFS-based algorithm to explore combinations of different entities. For any combination that could reformulate a new question, we allow it to generate a completely new question and store it for later evaluation. Specifically, in this step, we focus on the following processes:

(1) We use the DFS algorithm to select possible entity combinations. When the number of entities in a combination exceeds half of the filtered entities, we move on to step (2).

(2) We prompt the large language model with a concise zero-shot instruction to generate a structured statement containing all selected entities, based on the chosen entities and corresponding text. This statement serves as a conclusion that can be drawn from the document.

(3) After generating the statement, we ask the large language model to create a structured question that includes all the selected entities and can be answered, based on the generated statement.

(4) We then return the generated question to the large language model and ask it to verify whether the question includes all the necessary entities. If it does, we proceed to the next step (5); if not, we return to step (1) and select a new combination.

(5) If the question contains all the entities, we input it again into the large language model to check whether it can be answered. If it can, we store both the question and the count of its entities for final selection. If not, we discard it and return to step (1) to

Subset Name	Test Data Size	Document Length	Question Length	Entity Numbers	Domain
SQuADv2	507 / 1000	189.74 ± 77.59	13.85 ± 4.32	2.29 ± 0.95	Wikipedia
QA^2	247 / 506	1109.45 ± 899.83	10.29 ± 2.37	2.05 ± 0.57	Mostly Wikipedia
BanditQA	736 / 2070	363.67 ± 202.03	10.36 ± 3.16	1.70 ± 0.72	Wikipedia
BBC	59 / 278	652.90 ± 385.85	20.61 ± 4.73	3.31 ± 1.04	News
Reddit	113 / 313	569.48 ± 364.64	16.58 ± 3.49	2.93 ± 0.78	Social Media
Yelp	51 / 165	486.87 ± 184.44	17.72 ± 3.82	3.04 ± 0.76	Review

Table 1: The detail information of all experimental data for evaluating the performance on the question reformulation task. The *Length* is calculated based on the number of tokens after tokenization using GEMMA2-9B.

search for another combination. (Note: When the number of valid questions reaches the set threshold, no further combinations will be attempted.)

In this step, our algorithm uses a DFS search approach to identify possible entity combinations that can be reformulated into new questions. During the generation process, we find that directly prompting the large language model to reshape the question often results in a low success rate due to the model's potentially unlimited output content. To address this, we guide the large language model to generate structured outputs by restricting the entities required in the generated text. We ensure the reliability of the reformulated question by first generating a statement, then using that statement to create the question.

Additionally, to enhance the algorithm's efficiency, we verify the entities immediately after generating the new question, preventing unnecessary subsequent steps. We also apply pruning to the DFS algorithm by limiting the total number of preselected questions and controlling the search depth.

2.3 Candidate Question Re-evaluation

The DFS search process generates multiple candidate questions, each preserving different aspects of the original question. To select the optimal reformulation, we develop a two-stage re-evaluation strategy.

First, we assess each candidate's answerability by prompting the large language model to verify whether the question can be answered using only the information present in the document. This ensures that our final selection maintains semantic validity and practical utility.

Second, we consider the entity overlap score of each answerable candidate. Questions with higher entity overlap are preferred as they better maintain the user's original intent. This is quantified by:

$$\text{Entity Overlap Score} = \frac{|\text{Entities}_{\text{cand}} \cap \text{Entities}_{\text{orig}}|}{|\text{Entities}_{\text{orig}}|}$$

When multiple candidates achieve similar answerability, we select the one with the highest entity overlap score. As the example illustrated in Figure 2, we identify three questions that are considered potentially answerable by the large language model. Therefore, we allow the model to select the most optimal reformulated question. While both Q2 and Q4 are recognized as clearly answerable, Q4 is ultimately chosen due to its inclusion of a greater number of valid entities. Thus, the model returns Q4 as the final reformulated question to the user. This approach ensures that the final output: (1) can be answered from the given document, (2) maximally preserves the user's original intent, and (3) maintains semantic coherence with the source text.

3 Experiments

3.1 Datasets

In this paper, we utilize the newly constructed high-quality dataset **CouldAsk**[1], introduced by Zhao et al. (2024). This dataset comprises subsets from six diverse sources, including Yelp, BBC, SQuAD, Reddit, and others, featuring over 4,000 high-difficulty, high-quality data points filtered by both human annotators and GPT-4. For our study, we focus on over 1,700 unanswerable questions selected from all subsets as test data for model reformulation. The detailed information about our experimental data is provided in Table 1.

3.2 Large Language Models

To evaluate our methodology and compare it with baseline approaches from previous studies,

[1] https://huggingface.co/datasets/wentingzhao/couldask

Evaluator	QA^2	BanditQA	BBC	Reddit	Yelp	SQuADv2	Average
	200 / 247	300 / 507	50 / 59	100 / 113	50 / 51	300 / 736	1000 / 1713
LLAMA2-7B (Zhao et al., 2024)	39.00	68.67	14.00	79.00	44.00	72.00	52.78
GPT-4O-MINI	90.50	92.33	88.00	92.00	90.00	91.33	90.70

Table 2: The comparison of prediction accuracy between previous LLAMA2-7B evaluator proposed by Zhao et al. (2024) and our proposed GPT-4O-MINI evaluator.

we conduct all experiments using four different LLMs: GPT-3.5, GPT-4O-MINI, GEMMA2-9B and QWEN2.5-7B. The detailed information of different LLMs is shown in Appendix B.

3.3 Metric

We employ the metric **Accuracy**, in line with previous research, to measure the proportion of unanswerable questions successfully reformulated. This evaluation involves two criteria:

(i) We use GPT-4O-MINI as an evaluator to generate reasoning steps and assess whether the reformulated question can be answered using the relevant text. (ii) We use GPT-4O-MINI as an entity detector to measure the overlap of entities between the reformulated question and the original question. If the number of overlapping entities exceeds half of the labeled entities in the dataset, we consider the reformulated question likely to help the inquirer obtain the desired information.

A reformulated question is deemed successful and counted towards accuracy only if it satisfies both criteria.

4 GPT Evaluator

In the previous paper, Zhao et al. (2024) fine-tuned the LLAMA2-7B model (Touvron et al., 2023) on the training dataset for the sequence classification task to determine whether a question could be answered from the corresponding text. This approach achieved high scores on their validation dataset. However, in our experiments, we find that its generalization performance is suboptimal; even on the **CouldAsk** dataset, which includes training data, it still yields extremely poor classification results.

Given the situation described above, we find it necessary to propose a reliable and efficient evaluation method. Therefore, in this paper, we choose the GPT-4O-MINI model, which demonstrates excellent language understanding capabilities, as the evaluation model, ensuring both strong evaluation performance and cost-effectiveness.

To demonstrate the strong capabilities of the GPT-4O-MINI model, we select a subset of labeled questions from each portion of the **CouldAsk** dataset. We then have both evaluation models perform classification predictions and record their classification accuracy on this data, with the results shown in Table 2.

We observe that GPT-4O-MINI consistently achieves high scores across all tests, whereas the fine-tuned LLAMA2-7B model shows significantly lower accuracy than GPT-4O-MINI and exhibits substantial score variation across different datasets (ranging from a high of 79% to a low of 14%). For the average accuracy across six different datasets, our GPT-4O-MINI evaluator achieves over 90%, while the LLAMA2-7B evaluator only reaches 52%, performing slightly better than random predictions. This clearly indicates that the fine-tuned model lacks generalizability across diverse data. Therefore, using GPT-4O-MINI as the evaluator is not only more reliable but also provides faster evaluation and lower operational costs.

5 Experimental Results

To demonstrate the effectiveness of our proposed zero-shot DRS method, we test it on six datasets, totaling over 1,700 data points, using four different large language models. We conduct a fair and thorough comparison between our method and four baseline methods from previous research, with all accuracy scores presented in Table 3. Additionally, we adjust relevant parameters, such as temperature and the number of candidate questions, in the zero-shot DRS method and conduct multiple experiments. The results are shown in Figure 4 and Figure 3. Finally, considering GPU resources and API costs, we select a subset of test data for additional experiments with more models, as listed in Appendix D.

Effective DRS. Table 3 clearly demonstrates the strong capabilities of our proposed zero-shot DRS method. Our method significantly outperforms all

Model	Method	QA^2	BanditQA	BBC	Reddit	Yelp	SQuADv2	Average
GPT-3.5	Zero-Shot	28.74	11.41	13.56	18.58	19.61	36.69	21.43
	Zero-Shot w/ CoT	29.15	15.63	13.56	20.35	15.69	43.79	23.03
	Few-Shot	32.79	22.69	15.25	23.01	29.41	38.46	26.94
	Few-Shot w/ CoT	44.94	48.78	16.95	18.58	25.49	41.22	32.66
	Zero-Shot DRS (ours)	**81.80**	**73.20**	**62.71**	**75.22**	**66.67**	**62.90**	**70.42**
GPT-4O-MINI	Zero-Shot	36.44	15.08	15.25	23.01	15.69	47.93	25.57
	Zero-Shot w/ CoT	49.39	37.50	42.37	17.70	35.29	50.10	38.73
	Few-Shot	36.44	17.80	16.95	21.24	21.57	46.55	26.76
	Few-Shot w/ CoT	61.13	52.85	47.46	43.36	56.86	59.57	53.54
	Zero-Shot DRS (ours)	**88.26**	**80.16**	**79.66**	**83.19**	**78.43**	**78.30**	**81.33**
QWEN2.5-7B	Zero-Shot	46.15	24.05	25.42	23.89	25.49	46.70	31.95
	Zero-Shot w/ CoT	47.18	44.21	27.12	35.40	31.37	53.55	39.81
	Few-Shot	39.68	25.14	15.25	23.89	19.61	39.64	27.20
	Few-Shot w/ CoT	54.28	45.59	25.42	25.66	33.33	50.64	39.15
	Zero-Shot DRS (ours)	**77.73**	**69.84**	**55.93**	**53.10**	**58.82**	**60.55**	**62.66**
GEMMA2-9B	Zero-Shot	36.03	21.33	20.34	18.58	19.61	42.21	26.35
	Zero-Shot w/ CoT	29.15	17.12	22.03	23.01	19.61	32.94	23.96
	Few-Shot	26.32	16.17	25.42	15.04	17.65	29.39	21.67
	Few-Shot w/ CoT	22.67	18.07	10.17	24.78	25.49	27.42	21.43
	Zero-Shot DRS (ours)	**59.92**	**60.73**	**55.93**	**59.29**	**49.02**	**55.62**	**56.75**

Table 3: The main experimental results of four different large language models on full data across six different datasets, where best results are highlighted in **bold** font.

baselines across all datasets and models. Using the GPT-3.5 model, the average accuracy in the zero-shot setting increases from 23% to 70%, nearly tripling the original performance — a remarkable improvement that underscores the effectiveness of our approach. With the GEMMA2-9B model, while the results are lower than those of GPT-3.5, our method still boosts the zero-shot accuracy from 26% to 57%, achieving an improvement of over 100%. Moreover, even when compared to few-shot baselines, our zero-shot method consistently outperforms them. On GPT-3.5, the 70% average accuracy is more than double the 33% accuracy of the few-shot CoT method. For GEMMA2-9B, the few-shot accuracy is even lower than the zero-shot baseline.

In addition to the significant improvements observed with GEMMA2-9B and GPT-3.5, both GPT-4O-MINI and QWEN2.5-7B also show remarkable enhancements. For GPT-4O-MINI, the accuracy achieved by zero-shot DRS is more than double that of zero-shot CoT, and still shows a 52% improvement over the few-shot CoT results. While QWEN2.5-7B exhibits the smallest improvement ratio among the four models, it still achieves over a 57% enhancement compared to both zero-shot and few-shot CoT. These comparisons collectively demonstrate the effectiveness of the zero-shot DRS method in question reformulation.

Robust DRS. In the process of our DRS method, several steps involve using LLMs to generate corresponding answers. The model's temperature influences the output, affecting the diversity and reliability of the responses. Due to the continuity and interrelation of these steps, errors can propagate and accumulate. Therefore, we apply different temperatures to various LLMs in our method, and the experimental results are presented in Figure 4.

We observe that the average scores remain remarkably stable as the temperature increases. The score differences between different temperature settings are at most within 3 percentage points, which is negligible compared to the performance improvement brought by our method. Additionally, we notice that GPT-3.5 and GPT-4O-MINI achieve better scores when the temperature is set to 0.0 or 0.7. This is likely because, at a temperature of 0.0, the models can more strictly follow the instructions containing entities, while at 0.7, the diversity of the outputs is effectively utilized, providing more varied options for candidate questions and increasing the likelihood of selecting a better final question. In contrast, the accuracy of QWEN2.5-7B and GEMMA2-9B slightly decreases with higher tem-

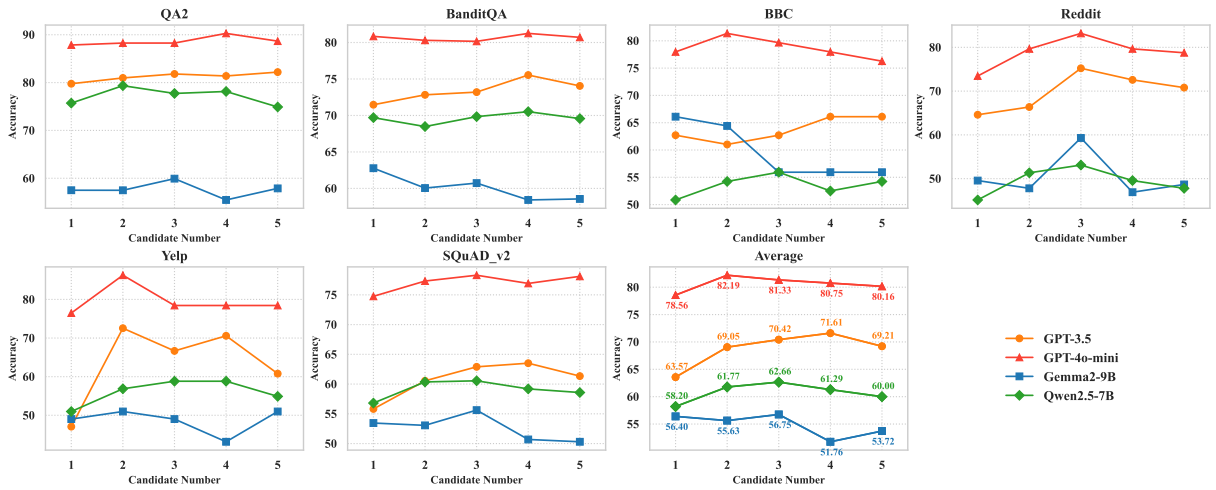

Figure 3: Model accuracy across four large language models with varying numbers of candidate questions, evaluated on six datasets and their average.

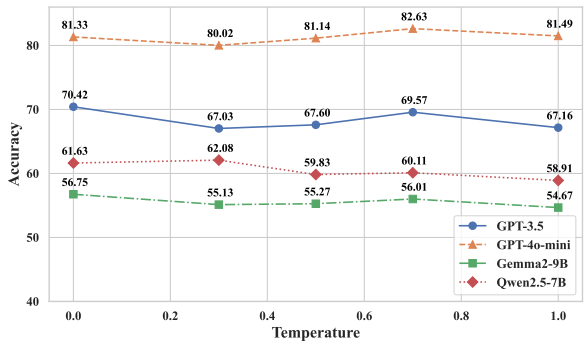

Figure 4: Model accuracy across four large language models with five different temperature settings, evaluated on six datasets and shows their average.

peratures, as smaller models struggle to follow instructions effectively as the temperature increases.

Impact of the Number of Candidate Questions. In our DRS method, before returning the final output question, we guide the model to generate a certain number of candidate questions. To explore the impact of this parameter on the performance of our method, we conduct experiments with varying numbers of candidate questions and observe how accuracy changes. We set the number of candidate questions from 1 to 5 for each large language model and test on all datasets, with the results shown in Figure 3.

We observe that the number of candidate questions does affect the final accuracy. Notably, when only one candidate question is generated, the accuracy of question reformulation across various models drops more significantly compared to other settings. For instance, with GPT-3.5, setting the number of candidate questions to 1 results in a decrease in accuracy from 70.42% to 63.57%, a drop of nearly 10%. Similarly, GPT-4O-MINI 's accuracy decreases from 81.33% to 78.56%. This is because, when only one question is generated, inaccuracies in entity extraction, coupled with a shallow search depth in our DRS algorithm, make it challenging to ensure significant overlap between the entities in the reformulated question and the true label entities. Consequently, the reformulated question often fails to meet the entity overlap requirement.

However, when the number of candidate questions is set from 2 to 5, the accuracy differences are relatively minor, typically within 3 percentage points (only QWEN2.5-7B shows a larger decrease when candidate number is 4), following a general trend of initially increasing and then decreasing. Besides, the curve shows that setting the number of candidate questions to 2 or 3 yields the most ideal results, achieving high accuracy while significantly saving search time. Even more convincing is the fact that, even when the number of candidate questions is set to the least ideal value of 1, the scores obtained by our DRS method still far outperform all other baselines. This strongly demonstrates the power and remarkable robustness of our proposed method.

6 Case Study

To highlight the advantages of our proposed zero-shot DRS method, we select some examples to demonstrate its effectiveness compared to other baselines, as shown in Figure 5. We observe that the relevant documents primarily discuss mathe-

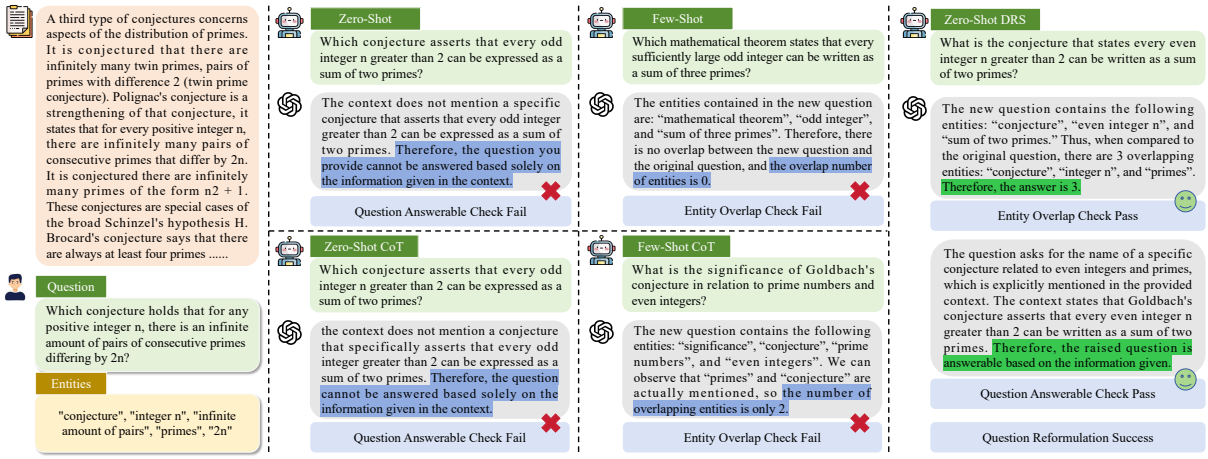

Figure 5: Comparison of our zero-shot DRS method with four baseline approaches, where our method successfully reformulates the question while all other baselines do not.

matical theorems, and the user's question involves five entities but cannot be answered using the corresponding documents. In these examples, the reformulated questions generated using zero-shot and zero-shot CoT are identical. These questions overlap with the original question on three entities: "conjecture", "integer n", and "primes". However, they still cannot be answered based on the provided documents, making such reformulations unsuccessful.

In contrast, questions generated using the few-shot or few-shot CoT methods are more diverse. However, despite being answerable based on the document, they fail to pass the entity overlap check. Specifically, the questions generated by the few-shot method contain no entities from the original question, while those generated by the few-shot CoT method include only two entities. Both fail to meet the required entity overlap ratio of 0.5 or higher, making these reformulations unsatisfactory.

In comparison, the reformulated question generated by our proposed DRS method satisfies both strict conditions: answerability and entity overlap ratio. The DRS method generates questions through a DFS search process, where the model is guided by the constraints of three entities: "conjecture", "integer n", and "primes". In this process, we first prompt the model to construct a statement based on the document, using these entities. We then use this statement to generate a reformulated question containing the same entities. This approach ensures both answerability and entity overlap, optimizing the model's ability to help users obtain relevant answers.

From these observations, we conclude that even advanced methods like few-shot or few-shot CoT struggle to produce satisfactory reformulations due to the lack of constraints on generated content. This lack of control often results in outputs that either overly focus on preserving the original question's entities, leading to unanswerable questions, or prioritize answerability at the expense of entity overlap. In contrast, the DRS method effectively balances these two requirements by systematically searching for entity combinations and generating structured statements and reformulated questions containing the specified entities. This approach unlocks the full potential of LLMs, enabling more effective question rephrasing for users.

7 Conclusion

In our paper, we propose a zero-shot DRS method that significantly enhances the performance of large language models on the unanswerable question reformulation task. Our method outperforms all other baselines, including the few-shot CoT approach, across six different datasets and various LLMs. We also conduct extensive experiments on key parameters, such as temperature and the number of candidate questions, with results demonstrating the strong capabilities and robustness of our approach. Furthermore, our findings reveal that the previously proposed LLAMA2-7B evaluator lacks the capability to fairly assess the reformulated questions. We propose using GPT-4O-MINI as a more reliable, accurate, and low-latency evaluator for future research. Looking ahead, we aim to develop even more effective methods to harness the full potential of LLMs, helping them assist people in understanding unfamiliar documents more effectively.

Limitation

In this paper, our proposed zero-shot DRS method significantly enhances the ability of LLMs to reformulate unanswerable questions. However, there are still the following limitations: (i) Despite the use of efficient pruning with the DFS algorithm, the process still requires multiple passes through the document, which increases computational costs. Future research could explore methods to enable LLMs to complete the reformulation in a single attempt. (ii) While we conduct tests on six datasets, they do not include documents from specific disciplines, which are common in real-world applications. Future work could focus on developing datasets that cover a broader range of domains, allowing for a more comprehensive evaluation of algorithmic and large language model capabilities.

Ethics Statement

Ethical considerations are of utmost importance in our research endeavors. In this paper, we strictly adhere to ethical principles by exclusively utilizing open-source datasets and employing various models that are either open-source or widely recognized in the scientific community. Our proposed methodology aims to enhance the model's ability to reformulate unanswerable questions when encountering documents from new knowledge domains in real-world scenarios. We are committed to upholding ethical standards throughout the research process, prioritizing transparency, and promoting the responsible use of technology for the betterment of society.

Acknowledgements

The authors would like to thank the anonymous reviewers for their discussion and feedback.

References

Akari Asai and Eunsol Choi. 2020. Challenges in information-seeking qa: Unanswerable questions and paragraph retrieval. *arXiv preprint arXiv:2010.11915*.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Yang Deng, Yong Zhao, Moxin Li, See-Kiong Ng, and Tat-Seng Chua. 2024. Gotcha! don't trick me with unanswerable questions! self-aligning large language models for responding to unknown questions. *arXiv preprint arXiv:2402.15062*.

Matthew Dunn, Levent Sagun, Mike Higgins, V. Ugur Guney, Volkan Cirik, and Kyunghyun Cho. 2017. Searchqa: A new q&a dataset augmented with context from a search engine.

Pedro Faustini, Zhiyu Chen, Besnik Fetahu, Oleg Rokhlenko, and Shervin Malmasi. 2023. Answering unanswered questions through semantic reformulations in spoken QA. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 5: Industry Track)*, pages 729–743, Toronto, Canada. Association for Computational Linguistics.

Ge Gao, Hung-Ting Chen, Yoav Artzi, and Eunsol Choi. 2023a. Continually improving extractive QA via human feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 406–423, Singapore. Association for Computational Linguistics.

Lingyu Gao, Aditi Chaudhary, Krishna Srinivasan, Kazuma Hashimoto, Karthik Raman, and Michael Bendersky. 2023b. Ambiguity-aware in-context learning with large language models. *arXiv preprint arXiv:2309.07900*.

Felix Hill, Antoine Bordes, Sumit Chopra, and Jason Weston. 2016. The goldilocks principle: Reading children's books with explicit memory representations.

Ziwei Ji, Tiezheng Yu, Yan Xu, Nayeon Lee, Etsuko Ishii, and Pascale Fung. 2023. Towards mitigating llm hallucination via self reflection. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 1827–1843.

Hyuhng Joon Kim, Youna Kim, Cheonbok Park, Junyeob Kim, Choonghyun Park, Kang Min Yoo, Sang goo Lee, and Taeuk Kim. 2024. Aligning language models to explicitly handle ambiguity.

Najoung Kim, Phu Mon Htut, Samuel R. Bowman, and Jackson Petty. 2023. $(QA)^2$: Question answering with questionable assumptions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8466–8487, Toronto, Canada. Association for Computational Linguistics.

Khalid Nassiri and Moulay Akhloufi. 2023. Transformer models used for text-based question answering systems. *Applied Intelligence*, 53(9):10602–10635.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, et al. 2024. Gpt-4 technical report.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback.

Raul Puri, Ryan Spring, Mostofa Patwary, Mohammad Shoeybi, and Bryan Catanzaro. 2020. Training question answering models from synthetic data. *arXiv preprint arXiv:2002.09599*.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad.

Elior Sulem, Jamaal Hay, and Dan Roth. 2021. Do we know what we don't know? studying unanswerable questions beyond squad 2.0. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4543–4548.

Elior Sulem, Jamaal Hay, and Dan Roth. 2022. Yes, no or idk: The challenge of unanswerable yes/no questions. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1075–1085.

Yiming Tan, Dehai Min, Yu Li, Wenbo Li, Nan Hu, Yongrui Chen, and Guilin Qi. 2023. Can chatgpt replace traditional kbqa models? an in-depth analysis of the question answering performance of the gpt llm family. In *International Semantic Web Conference*, pages 348–367. Springer.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, Anton Tsitsulin, Nino Vieillard, et al. 2024. Gemma 2: Improving open language models at a practical size.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

SM Tonmoy, SM Zaman, Vinija Jain, Anku Rani, Vipula Rawte, Aman Chadha, and Amitava Das. 2024. A comprehensive survey of hallucination mitigation techniques in large language models. *arXiv preprint arXiv:2401.01313*.

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

Zhen Wang. 2022. Modern question answering datasets and benchmarks: A survey.

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

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zhihao Fan. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Zhangyue Yin, Qiushi Sun, Qipeng Guo, Jiawen Wu, Xipeng Qiu, and Xuanjing Huang. 2023. Do large language models know what they don't know?

Xinyan Yu, Sewon Min, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. CREPE: Open-domain question answering with false presuppositions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10457–10480, Toronto, Canada. Association for Computational Linguistics.

Michael JQ Zhang and Eunsol Choi. 2023. Clarify when necessary: Resolving ambiguity through interaction with lms. *arXiv preprint arXiv:2311.09469*.

Wenting Zhao, Ge Gao, Claire Cardie, and Alexander M. Rush. 2024. I could've asked that: Reformulating unanswerable questions.

A Implementation Details

We use the *gpt-3.5-turbo-0125* version for GPT-3.5, *gpt-4-0125-preview* for GPT-4, and *gpt-4o-mini* for GPT-4O-MINI. All experiments were conducted between November 9th and 25th, 2024.

To ensure fair experimental comparisons and re-evaluate the baseline scores reported in the previous paper, we use the open-source GitHub code provided by Zhao et al. (2024), maintaining consistency in the prompts and LLM parameters. The versions of all OpenAI models also remain consistent with the ones mentioned above.

In our main experiments using the proposed DRS method, we set the temperature to 0.0 for all large language models to ensure reproducibility. Similarly, for the GPT-4O-MINI evaluator, we set the *temperature* to 0.0 and *top_p* to 1.0 to maintain reliability and consistency in the model's judgments. For the two GPT series models, we use the official OpenAI API[2] for inference.

For the two open-source models GEMMA2-9B and QWEN2.5-7B, we use the weights released on HuggingFace[3] and deploy them on a single NVIDIA RTX A6000 GPU using **bfloat16** precision and utilize the generate function for text generation.

B Large Language Models

Closed-Source Models. To balance experimental cost and effectiveness, we primarily use two widely adopted models from OpenAI: GPT-3.5 and GPT-4O-MINI. In Appendix D, we select a subset for the experiment using GPT-4.

- GPT-3.5: A robust large language model developed by OpenAI, capable of generating text based on instructions, and highly effective across diverse natural language processing tasks.
- GPT-4: An advanced multi-modal language model from OpenAI that accepts both image and text inputs for text generation, achieving near-human performance on various benchmarks.
- GPT-4O-MINI: A cost-efficient multi-modal model released by OpenAI on July 18, 2024, offering low latency and cost while supporting a wide range of tasks.

[2] https://openai.com/
[3] https://huggingface.co/

Open-Source Models. We also perform experiments using two well-known models developed by Google and Alibaba: GEMMA2-9B (Team et al., 2024) and QWEN2.5-7B (Yang et al., 2024; Team, 2024). In Appendix D, we select a subset for the experiment using LLAMA3.1-70B and GEMMA2-27B.

- LLAMA3.1-70B: The 70B-parameter flagship model from the latest LLAMA3.1 open-source family released by MetaAI.
- GEMMMA2: The next-generation open-source model from Google, released on June 27, 2024, as an improved version of GEMMA, available in 2B, 9B, and 27B parameter configurations.
- QWEN2.5-7B: A 7B-parameter model from Alibaba's latest QWEN2.5 series, delivering strong performance on various benchmarks compared to other models of similar size.

C Related Work

Question answering has been a central focus in natural language processing (NLP) (Wang, 2022; Puri et al., 2020; Nassiri and Akhloufi, 2023), with the development of datasets like CBT and SearchQA (Hill et al., 2016; Dunn et al., 2017) designed to assess a model's ability to answer questions. However, these datasets do not address the challenge of unanswerable questions.

With the rise of GPT-3 (Brown et al., 2020) and even more powerful large language models (LLMs), the focus has shifted to handling unanswerable questions, which arise due to ambiguity or gaps in knowledge (Deng et al., 2024; Gao et al., 2023b; Yin et al., 2023).

Several studies have explored methods for resolving ambiguities in questions, often targeting issues like missing qualifiers that can lead to different interpretations (Kim et al., 2024; Zhang and Choi, 2023). Many research on questions outside the model's knowledge scope has focused on guiding models to recognize when they lack sufficient information to answer, thereby reducing hallucinations (Yin et al., 2023; Ji et al., 2023; Tonmoy et al., 2024). However, a commonly overlooked real-world scenario involves individuals posing seemingly relevant questions that are unanswerable due to their limited familiarity with the document's knowledge domain. In such cases, large

Model	Method	QA^2	BanditQA	BBC	Reddit	Yelp	SQuADv2	Average
GPT-4	Zero-Shot	44.00	30.00	28.81	25.66	21.57	64.00	35.67
	Few-Shot	48.00	29.33	37.29	23.01	35.29	60.67	38.93
	Zero-Shot DRS (ours)	**78.67**	**72.67**	**71.19**	**66.37**	**62.75**	**70.00**	**70.28**
LLAMA3.1-70B	Zero-Shot	36.00	31.33	33.90	19.47	21.58	58.00	33.38
	Few-Shot	29.33	29.33	27.12	18.58	25.49	36.67	27.75
	Zero-Shot DRS (ours)	**65.33**	**74.67**	**67.80**	**61.95**	**74.51**	**66.67**	**68.48**
GEMMA2-27B	Zero-Shot	10.00	12.00	23.73	18.58	17.65	42.67	20.77
	Few-Shot	28.67	8.00	6.78	4.42	13.73	20.67	13.71
	Zero-Shot DRS (ours)	**42.00**	**64.00**	**52.54**	**46.02**	**37.25**	**55.33**	**49.52**

Table 4: The additional experimental results of three different large language models on full data across six different datasets, where best results are highlighted in **bold** font.

language models can be employed to reformulate these questions, enabling the inquirers to obtain the information they likely intended to seek.

The issue of document-related unanswerable questions was first systematically addressed in a dataset by Rajpurkar et al. (2018), and later expanded by studies such as Yu et al. (2023), which analyzed Google search queries, and Kim et al. (2023), which examined Reddit discussions. Both studies focused on identifying questions with incorrect assumptions or presuppositions that made them unanswerable.

In recent work, Zhao et al. (2024) introduced a high-quality dataset of document-related unanswerable questions, exploring how large language models can assist in reformulating such questions. They evaluated several well-known large language models using basic approaches like zero-shot and more advanced ones such as few-shot or Chain of Thought (CoT) prompting (Wei et al., 2023), but the results were suboptimal. In this paper, we propose the zero-shot DRS method to significantly improve LLMs' ability to help users reformulate unanswerable questions, advancing progress in this area.

D Additional Experiments

To address the constraints of GPU resources and API costs, we randomly select appropriately sized subsets from each test dataset for experiments with three well-known models: GPT-4, LLAMA3.1-70B, and GEMMA2-27B, which demonstrates the generalizability of our approach. We use the complete datasets for Yelp, BBC, and Reddit, while for the three larger datasets, we randomly select 150 samples for the experiments. For comparison, we focus on the results of our DRS method against two representative baselines, zero-shot and few-shot, due to the high computational cost of the models. The experimental results appear in Table 4.

Based on the experimental results, our proposed DRS method demonstrates significant improvements over both zero-shot and few-shot baselines across three newly tested large language models. For instance, on LLAMA3.1-70B, zero-shot DRS achieves over a 100% improvement in accuracy compared to the baselines. Similarly, on GPT-4, the improvement approaches 100%. Furthermore, with the DRS method, the average reformulation success rate for these two models reaches nearly 70%, which is remarkable. In contrast, GEMMA2-27B achieves an average accuracy of only about 50% with DRS, lagging behind the other two models. This discrepancy may be attributed to differences in model size and instruction adherence. Nevertheless, on GEMMA2-27B, DRS still boosts accuracy by approximately 125% over the zero-shot baseline and by around 275% over the few-shot baseline. Overall, these results further validate the effectiveness of our DRS method, highlighting its strong generalization and robustness across diverse large language models.

E Inference Time of DRS

In Table 5, we present the average runtime of GPT-3.5 and GEMMA2-9B under different methods. For our DRS method, we include the runtime when the number of candidate questions is set to 2 or 3, as Figure 3 shows that optimal performance can already be achieved with 2 or 3 candidates. From

Model	Method	Inference Time
GPT-3.5	Zero Shot w/ CoT	6.80s
	Few Shot w/ CoT	7.32s
	DRS (candidate num = 2)	10.07s
	DRS (candidate num = 3)	11.44s
GEMMA2-9B	Zero Shot w/ CoT	12.51s
	Few Shot w/ CoT	13.82s
	DRS (candidate num = 2)	17.36s
	DRS (candidate num = 3)	19.65s

Table 5: Average inference time per sample across six datasets for three types of methods.

Dataset	Human Evaluation
QA^2	100 / 100 (100%)
BanditQA	100 / 100 (100%)
BBC	59 / 59 (100%)
Reddit	100 / 100 (100%)
Yelp	51 / 51 (100%)
SQuADv2	99 / 100 (99%)

Table 6: Human evaluation results on six datasets for assessing the meaningfulness and relevance of reformulated questions.

Table 5, we observe that the DRS method, implemented based on a DFS search approach, results in longer runtimes compared to the few-shot CoT method, despite the use of effective pruning techniques. Specifically, when the number of candidate questions is 2, the runtime increases by 37% on GPT-3.5 and 25% on GEMMA2-9B.

However, despite the slight increase in inference time, the accuracy improvements of the DRS method over the few-shot CoT method are remarkable, with increases of 113% (from 32.66 to 69.55) and 160% (21.43 to 55.63) on the listed two large language models, respectively. This clearly shows a trade-off between runtime and accuracy, but the accuracy gains of the DRS method far outweigh the impact of the longer runtime. Therefore, we have strong evidence to support that the DRS method is a reasonable and efficient approach.

F Human Evaluation

In this paper, we propose a zero-shot DRS method to significantly improve the question reformulation task. However, a key concern arises: *Do LLMs merely reformulate extremely simple questions that are unrelated to the passages but can be easily answered by the models themselves?* For example, *How to spell wasabi?*

To thoroughly assess the effectiveness of the DRS method, we recruit three native English-speaking graduate students to evaluate all datasets. For subsets with over 100 samples, we randomly selected 100 reformulated questions for evaluation. A question is deemed valid if at least two students agree that it is meaningful and relevant to the passage. The human evaluation results are presented in Table 6.

The results clearly show that our proposed zero-shot DRS method generates reformulated questions that are nearly 100% meaningful and highly relevant to the given long context. Only one reformulated question within the SQuADv2 subset is deemed unmeaningful by three evaluators. This highlights the effectiveness of our method in faithfully preserving the user's original intent and producing relevant reformulated questions.

G Experimental Prompts

Entity Extraction Prompt

Find out all entities in the following question: {question}. The entities you find must be exactly exist in the given question. You should only return the entities, separated by comma and space.

Figure 6: Entity Extraction Prompt

Entity Categorize Prompt

Here is a question: {question}.
Here is an entity in this question: {entity}.
Tell me which category does this entity belong to in the given question - subject, object, predicate, attribute or others. You should only consider the situation of given entity in this specific question. Give your analysis within <analysis> tags, and only return its category name within <answer> tags.

Figure 7: Entity Classification Prompt

Statement and Question Generation Prompt

According to the following text: {context}. Generate a statement you could get from the given text that contains all following entities: {entities}. Then you are required to generate a question contains all given entities, which could be answered from your statement. Return the statement within <statement> tags and the question within <question> tags.

Figure 8: Statement and Question Generation Prompt

Entity Overlap Evaluation Prompt

Here is an original question: {question}, it contains the following entities: {entities}.
Here is a new question: {new_question}.
Tell me the number of overlapping entities between the new question and the original question, they do not need to be strictly the same, as long as mentioned, uppercase or lowercase doesn't matter. Give your analysis within <analysis> tag, and only return the math number of overlap entities within <answer> tags.

Figure 9: Entity Overlap Evaluation Prompt

Question Answerability Evaluation Prompt

Here is a long context: {context}.
Here is a question: {question}.
Tell me whether this question is answerable only according to the information in the provided context. Think carefully and give you analysis within <analysis> tags, then return only the final answer 'yes' or 'no' within <answer> tags.

Figure 10: Question Answerability Evaluation Prompt