

# Diversify-verify-adapt: Efficient and Robust Retrieval-Augmented Ambiguous Question Answering

Yeonjun In<sup>1\*</sup>, Sungchul Kim<sup>2†</sup>, Ryan A. Rossi<sup>2</sup>, Md Mehrab Tanjim<sup>2</sup>,  
Tong Yu<sup>2</sup>, Ritwik Sinha<sup>2</sup>, Chanyoung Park<sup>1</sup>

<sup>1</sup>KAIST <sup>2</sup>Adobe Research

{yeonjun.in, cy.park}@kaist.ac.kr

{sukim, ryrossi, tanjim, tyu, risinha}@adobe.com

## Abstract

The retrieval augmented generation (RAG) framework addresses an ambiguity in user queries in QA systems by retrieving passages that cover all plausible interpretations and generating comprehensive responses based on the passages. However, our preliminary studies reveal that a single retrieval process often suffers from low-quality results, as the retrieved passages frequently fail to capture all plausible interpretations. Although the iterative RAG approach has been proposed to address this problem, it comes at the cost of significantly reduced efficiency. To address these issues, we propose the **diversify-verify-adapt** (DIVA) framework. DIVA first **diversifies** the retrieved passages to encompass diverse interpretations. Subsequently, DIVA **verifies** the quality of the passages and **adapts** the most suitable approach tailored to their quality. This approach improves the QA systems' accuracy and robustness by handling low quality retrieval issue in ambiguous questions, while enhancing efficiency.

## 1 Introduction

Open-domain question answering (QA) systems aim to provide factual responses across diverse topics. However, ambiguity in user queries is common, with over 50% of Google search queries falling into this category (Min et al., 2020). Ambiguous questions challenge QA systems to determine user intent, making it essential for them to deliver answers covering all possible interpretations.

Addressing ambiguous questions is crucial in real-world applications, yet remains underexplored compared to unambiguous questions (Joshi et al., 2017; Kwiatkowski et al., 2019). This work aims to fill this gap by tackling the complexities of ambiguous QA.

Retrieval-augmented generation (RAG) framework has made significant progress in open-domain

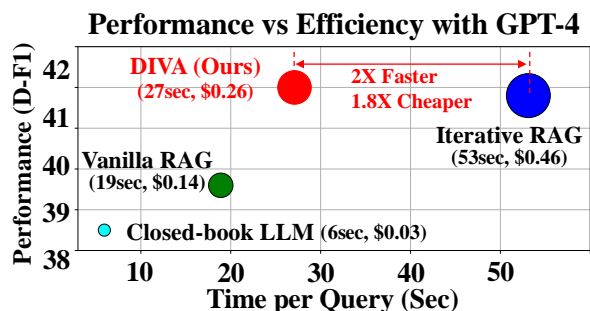


Figure 1: Trade-off between performance and efficiency under GPT-4 backbone on ASQA. Notably, DIVA achieves better performance to the iterative RAG (Kim et al., 2023), while significantly more efficient (that is, 2x faster and 1.8x cheaper). The size of the circle indicates the cost per query (\$). Closed-book LLM indicates the traditional few-shot prompting method used in Brown (2020)

QA tasks (Izacard and Grave, 2021; Lazaridou et al., 2022; Shi et al., 2023; Ram et al., 2023) and also proven to be an effective solution for addressing ambiguous questions (Min et al., 2020, 2021; Kim et al., 2023; Sun et al., 2023). Specifically, these approaches first retrieve passages on the given question and prompt the LLM to extract plausible interpretations and answers relying on the passages (c.f. Fig 2(a)).

Despite the success of the RAG framework on the ambiguous QA task, we should rethink: Is a single retrieval process sufficient to retrieve passages encompassing all plausible interpretations? To answer this question, we conduct preliminary experiments (c.f. Sec 2) about the quality of the retrieved passages used in the RAG framework. We observe that the passages obtained from the single retrieval process often pose a low quality issue with respect to addressing ambiguous questions. In other words, the retrieved passages often partially or completely failed to cover all plausible interpretations, leading to significant performance degradation in terms of factual accuracy.

To address this issue, the iterative RAG approach, ToC (Kim et al., 2023), has been introduced

\*Work done during internship at Adobe Research.

†Corresponding author.

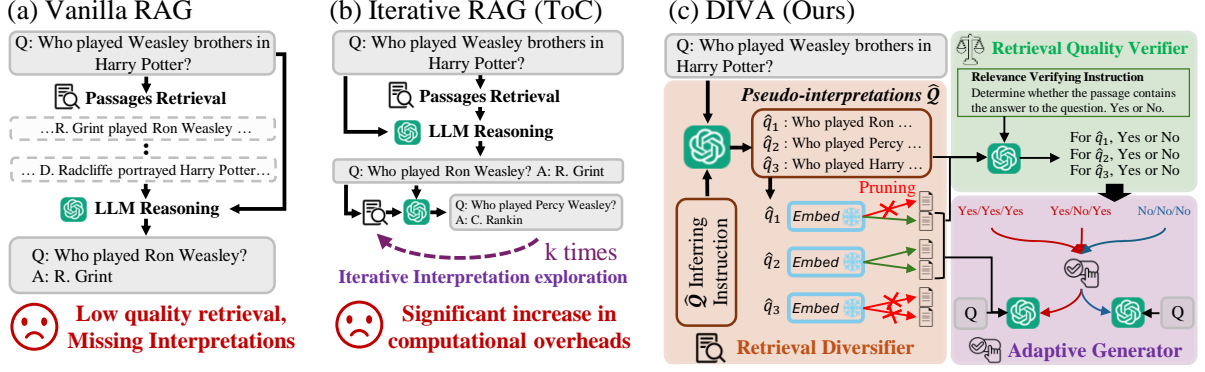


Figure 2: A conceptual comparison of RAG approaches to ambiguous QA. (a) Vanilla RAG retrieves passages and generates answers in a single pass, but it may not collect enough information for diverse interpretations (i.e., low-quality retrieval), compromising factual accuracy. (b) Iterative RAG retrieves passages and generates answers in a loop, using previous interpretations to enhance each subsequent iteration’s retrieval for exploring missing interpretations. Although effective, it is inefficient due to the repeated use of LLMs and retrievers. (c) DIVA retrieves passages covering diverse interpretations without relying on the iterative process and selects the most suitable knowledge for response generation by verifying retrieval quality.

(c.f. Fig 2(b)) to further explore other interpretations that can not be covered by the single retrieval process. Specifically, to further explore missing interpretations, the interpretations extracted in the previous iteration are utilized as queries to retrieve new passages, additional interpretations are then extracted. This exploration process is repeated in multiple times, leading to encompassing more diverse interpretations and corresponding answers. However, we argue that this effectiveness comes with a significant increase in computational overheads due to the iterative passage retrieval and LLM reasoning. In our experiments, this method requires an average of 5.5 exploration steps per query. As shown in Figure 1, Iterative RAG (i.e., ToC) significantly outperforms the vanilla RAG approach in terms of factual accuracy but at the cost of greatly reduced efficiency, with notable increases in both inference time and API call costs.

To this end, we introduce an efficient and robust RAG framework for ambiguous QA, referred to as **diversify-verify-adapt** (DIVA). DIVA comprises two key components efficiently addressing the low quality retrieval issue: **1) Retrieval Diversification (RD)** and **2) Adaptive Generation (AG)**. The key idea of RD is to infer *pseudo-interpretations* of a question, using them to retrieve a set of passages that broadly cover these interpretations, thus enhancing retrieval quality without any iterative interpretation exploration process. To further enhance the robustness of this framework, we propose an adaptive generation (AG) method. The key idea of AG is to carefully verify the overall quality of the passages retrieved from RD before indiscriminately incorporating

them. More specifically, we define a new criterion of quality levels tailored to ambiguous questions: {Useful, PartialUseful, Useless}. Subsequently, AG adapts the most suitable approach between relying on the retrieved passages and LLM’s internal knowledge, each of which is tailored to the specific quality level of the passages.

Experiments demonstrate that the proposed RD method efficiently diversifies the retrieval process to obtain passages covering diverse interpretations, thereby enhancing both QA and retrieval accuracy. Additionally, the proposed AG method successfully discriminate low quality passages, leading to the improvement of the QA performance. Consequently, DIVA outperforms existing baselines on the ASQA (Stelmakh et al., 2022) and SituatedQA (Zhang and Choi, 2021) across various LLM backbones in a few-shot setup, achieving superior accuracy and efficiency. The key contributions of this work are as follows:

- To the best of our knowledge, this paper is the first attempt to investigate the practical limitations of the existing RAG frameworks when applied to ambiguous QA task: low quality retrieval and inefficiency.
- We propose DIVA, an efficient and robust RAG framework that efficiently retrieves diverse passages, verifies their quality, and adapts the most suitable approach tailored to each retrieval quality.
- DIVA consistently outperforms state-of-the-art RAG approaches in ambiguous QA task, while significantly more efficient (nearly 1.5 - 3 times faster response generation).

## 2 Preliminary Experiments

We investigate the quality of retrieved passages and their impact on the performance of the RAG framework (as in Fig 2(a)) in ambiguous QA task.

**Experimental Details.** We utilize the most recent ambiguous QA dataset, ASQA (Stelmakh et al., 2022). We classify the quality of retrieved passages into three labels: **1) Fully Cover**, **2) Partially Cover**, and **3) Not Cover**. Fully Cover indicates that the retrieved passages encompass all plausible interpretations, Not Cover does that the retrieved passages do not contain any of them, and otherwise Partially Cover. We obtain these labels for each question by computing a string exact match between a set of retrieved passages and all plausible answers provided in ASQA as ground-truth answers. For implementation details of retrieving passages, see Appendix A.4.1.

**Results.** In Fig 3(a), we observe that for only 34.6% of questions (i.e., Fully Cover) the retriever successfully retrieves passages that cover all plausible interpretations. Additionally, for 15.7% of questions (i.e., Not Cover) the retriever fails to retrieve any relevant passages. More critically, as shown in Fig 3(b), the performance of the RAG framework (i.e., RAG in the figure) significantly deteriorates in terms of the factual accuracy (i.e., D-F1) when the retrieved passages pose a low quality issue (i.e., Partial Cover and Not Cover), indicating that it is highly susceptible to noise and irrelevant information in the ambiguous QA.

This observation raises a follow-up question: How can we handle cases where the retrieved passages do not fully cover the plausible answers? To address this issue, we conducted another experiment that compares the effectiveness of LLM’s internal knowledge and provided passages for different cases, respectively. We observe that when the retrieved passages do not contain any of the plausible interpretations (i.e., Not Cover), the closed-book LLM (i.e., LLM in the figure) significantly outperforms the RAG framework. This suggests that QA performance benefits more from relying on the LLMs’ internal knowledge rather than on external passages containing entirely irrelevant information.

In short, while the quality of retrieval is crucial for the performance of the RAG framework in ambiguous QA, existing works have largely overlooked this critical issue, which notably diminishes their practical applicability.

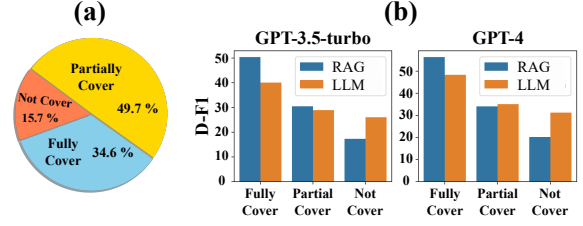


Figure 3: Preliminary results on ASQA. (a) Portion of each quality label of retrieved passages. (b) Performance comparison upon the quality label.

## 3 Proposed Method: DIVA

Based on the findings, we propose an efficient and robust RAG framework for ambiguous QA, **diversify-verify-adapt** (DIVA). This framework comprises two key components: Retrieval Diversification (Sec 3.2) and Adaptive Generation (Sec 3.3). The retrieval diversification method aims to efficiently **diversify** the retrieved passages to encompass diverse interpretations. Subsequently, the adaptive generation method aims to **verify** the quality of the passages and **adapt** the most suitable approach tailored to their quality. Fig 2(c) and Algorithm 1 show the overview and inference algorithm of DIVA, respectively.

### 3.1 Problem Formulation

Given an ambiguous question  $q_i$ , the goal of the proposed RAG framework is to generate a comprehensive response  $r_i$  that encompasses all plausible answers  $\mathcal{A}_i = \{a_{i,1}, \dots, a_{i,M}\}$  of the interpretations  $\mathcal{Q}_i = \{q_{i,1}, \dots, q_{i,M}\}$  based on the retrieved passages  $\mathcal{P}_i = \{p_{i,1}, \dots, p_{i,K}\}$ , where  $M$  and  $K$  indicate the number of plausible answers and passages, respectively. Specifically, given the  $\mathcal{P}_i$  ideally contains all  $\mathcal{Q}_i$  and  $\mathcal{A}_i$ , an LLM is first prompted with the question and the relevant passages to extract all plausible interpretations and their corresponding answers, formally represented as follows:

$$\mathcal{Q}_i, \mathcal{A}_i \leftarrow \text{LLM}(q_i, \mathcal{P}_i, I_e), \quad (1)$$

where  $I_e$  is a text prompt for extracting  $\mathcal{Q}_i$  and  $\mathcal{A}_i$  from the  $\mathcal{P}_i$ . Subsequently, based on the  $\mathcal{Q}_i$  and  $\mathcal{A}_i$ , the LLM is prompted to consolidate them with  $q_i$  and  $\mathcal{P}_i$  to generate a response  $r_i$ , formally represented as follows:

$$r_i \leftarrow \text{LLM}(\mathcal{Q}_i, \mathcal{A}_i, \mathcal{P}_i, q_i, I_g). \quad (2)$$

For the prompts  $I_e$  and  $I_g$ , we start with that of Kim et al. (2023) and modify it for our setup (see Table 10 and Table 11 in Appendix D).

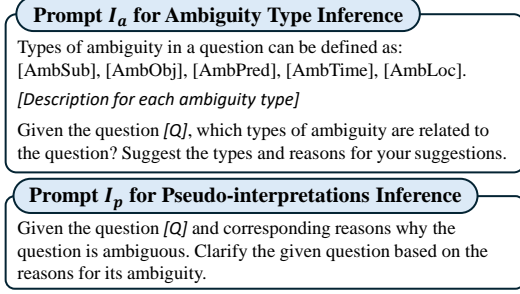


Figure 4: Conceptual example of prompts for pseudo-interpretations inference.

### 3.2 Retrieval Diversification (RD)

In this section, we propose a novel retrieval diversification (RD) method aiming to efficiently identify passages  $\mathcal{P}_i$  encompassing all plausible answers  $\mathcal{A}_i$  of the interpretations  $\mathcal{Q}_i$ . The key idea of RD is to infer *pseudo-interpretations*<sup>1</sup> of a question, using them to retrieve a set of passages that maximally cover these interpretations. This approach guarantees the retrieved passages encompass diverse interpretations, without any iterative interpretation exploration process of Kim et al. (2023), leading to the generated response  $r_i$  covering all  $\mathcal{A}_i$ .

**Inferring Pseudo-Interpretations.** To infer *pseudo-interpretations*  $\hat{\mathcal{Q}}_i = \{\hat{q}_{i,1}, \hat{q}_{i,2}, \dots\}$ , each of which related to a true plausible answer of  $\mathcal{A}_i$ , we draw inspiration from a human’s reasoning chain inferring multiple interpretations of a question. Given an ambiguous question, a human would first identify the ambiguous part of the question and then determine the reason for the ambiguity, followed by inferring multiple interpretations of the question. For example, given the question "Who played the Weasley brothers in Harry Potter?", the ambiguous part is the object of the question, "Weasley brothers," and the corresponding reason is that "It can refer to multiple characters such as Ron, Percy, and so on." Consequently, a human would generate "Who played Ron Weasley in Harry Potter?", "Who played Percy Weasley in Harry Potter?", etc.

To mimic this reasoning chain, we leverage the LLM’s reasoning ability to 1) identify the ambiguous part of the question and the reason for the ambiguity and 2) infer the *pseudo-interpretations*  $\hat{\mathcal{Q}}_i$  from the results. But, handling both tasks simultaneously would place a substantial load on a single LLM (See Appendix B.1 for a detailed discussion). As a result, we assign each task to a different LLM, formally represented as:

$$\hat{\mathcal{Q}}_i \leftarrow \text{LLM}(q_i, I_p, \text{LLM}(q_i, I_a)), \quad (3)$$

where  $I_p$  and  $I_a$  are carefully designed instructions for each step, respectively. We present the conceptual example of  $I_a$  and  $I_p$  in Fig 4 and full instructions in Table 7 and 8 of Appendix D. For  $\text{LLM}(\cdot)$ , we consider GPT-3.5 (Brown, 2020) and GPT-4 (Achiam et al., 2023).

**Retrieving Relevant and Diverse Passages.** As a first stage retrieval, we obtain the candidate passages  $\mathcal{C}_i$  generally relevant to the given question  $q_i$  from Wikipedia<sup>2</sup>. From the  $\mathcal{C}_i$ , we select a set of multiple passages  $\mathcal{P}_i$  with maximal coverage of all distinct *pseudo-interpretations*  $\hat{\mathcal{Q}}_i$ .

Retrieval for unambiguous questions involves scoring a **single passage** individually based on their relevance to a **single interpretation**. Whereas, when it comes to the ambiguous questions, we should retrieve a **set of passages** encompassing **multiple interpretations**, which makes this problem more challenging. To obtain such set of passages, we explicitly employ our inferred *pseudo-interpretations*  $\hat{\mathcal{Q}}_i$  to retrieve the set of passages  $\tilde{\mathcal{P}}_i$  that maximally cover these interpretations, formally represented as follows:

$$\tilde{\mathcal{P}}_i \leftarrow \bigcup_{j=1}^{|\hat{\mathcal{Q}}_i|} \mathcal{R}(\mathcal{C}_i, \hat{q}_{i,j}; K), \quad (4)$$

where  $\mathcal{R}$  is a retriever yielding top- $K$  passages from the  $\mathcal{C}_i$  by relevance scores to each *pseudo interpretation*  $\hat{q}_{i,j}$ .

**Pruning Noisy Passages.** Although this process explicitly enables  $\tilde{\mathcal{P}}_i$  to encompass all *pseudo interpretations*, there could be some noisy and irrelevant passages due to the absence of perfect retriever and the noise of the inferred *pseudo-interpretations*  $\hat{\mathcal{Q}}_i$ . To this end, we find and prune the passages that are highly likely to be irrelevant and noisy. Our intuition is that 1) noisy passages caused by the imperfect retriever tend to be irrelevant to all *pseudo-interpretations* and 2) noisy passages caused by noisy *pseudo-interpretations* tend to be irrelevant to most of the *pseudo-interpretations*. Based upon this intuition, we measure an averaged relevance of the passage to determine if it is noisy or not. The averaged relevance of a passage  $\mathcal{S}(p)$  is calculated as follows:

$$\mathcal{S}(p) \leftarrow \frac{1}{|\hat{\mathcal{Q}}_i|} \sum_{j=1}^{|\hat{\mathcal{Q}}_i|} \frac{\text{Enc}(\hat{q}_j) \cdot \text{Enc}(p)}{\|\text{Enc}(\hat{q}_j)\| \cdot \|\text{Enc}(p)\|}, \quad (5)$$

<sup>1</sup>We define *pseudo-interpretations* as approximate interpretations closely resembling the actual interpretations.

<sup>2</sup>We use ColBERT (Khattab and Zaharia, 2020) and Bing search API as retrievers.



where  $\text{Enc}(\cdot)$  encodes sentences to a dense vector and  $p \in \tilde{\mathcal{P}}_i$ . We then select the top- $K$  passages from the  $\tilde{\mathcal{P}}_i$  based on these averaged scores as the final passage set  $\mathcal{P}_i$ .

Our approach is generic, allowing for the use of various sentence embedding models for calculating relevance scores. In line with the sota baseline (Kim et al., 2023), we employ the frozen Sentence-BERT (Reimers and Gurevych, 2019) in  $\mathcal{R}(\cdot)$  and  $\text{Enc}(\cdot)$  in our implementation.

### 3.3 Adaptive Generation (AG)

Despite the effectiveness of the proposed RD method, there may be the low quality of  $\mathcal{P}_i$ . To further enhance the robustness of DIVA, in this section, we propose an adaptive generation method. The key idea of AG is to carefully verify the overall quality of the passages retrieved from RD before indiscriminately incorporating them.

From the findings in Section 2, if  $\mathcal{P}_i$  does not encompass all plausible interpretations,  $\mathcal{Q}_i$  and  $\mathcal{A}_i$ , the response generated by the RAG framework is highly likely to be inaccurate. To this end, we introduce an adaptive generation (AG) method that dynamically adjust the response generation strategy among the RAG framework and closed-book LLM, which is achieved by verifying the quality of  $\mathcal{P}_i$  before attempting a solution.

**Retrieval Verification (RV)** To verify the quality of  $\mathcal{P}_i$ , we exploit the LLM’s strong natural language understanding ability. The existing works (Li et al., 2023; Asai et al., 2023; Yan et al., 2024) verify whether  $\mathcal{P}_i$  can sufficiently support answering  $q_i$  by prompting or training the LLM to give a proper label  $V_i$  (e.g., Yes / No):

$$V_i \leftarrow \text{LLM}(q_i, \mathcal{P}_i, I_v), \quad (6)$$

where  $I_v$  is the corresponding instruction. However, the retrieval quality in terms of ambiguous questions should be graded according to how many interpretations are encompassed by the retrieved passages, which can not be achieved by the existing approaches tailored to unambiguous questions. To this end, we newly define a criterion of quality levels tailored to ambiguous questions: {Useful, PartialUseful, Useless}. Useful indicates the  $\mathcal{P}_i$  encompasses all  $\mathcal{Q}_i$  and  $\mathcal{A}_i$ , Useless indicates the  $\mathcal{P}_i$  does not contain any of them, otherwise PartialUseful. To determine these grades, we estimate how many interpretations are encompassed by the  $\mathcal{P}_i$  by explicitly utilizing the *pseudo-interpretations*  $\hat{\mathcal{Q}}_i$ :

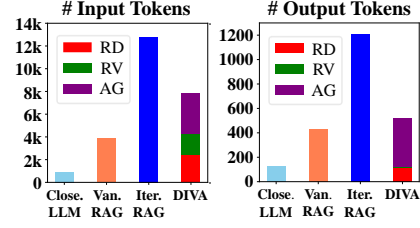


Figure 5: Comparison of the number of tokens per query using GPT-4 backbone. RD, RV, and AG indicate the proposed retrieval diversify, retrieval verify, and adaptive generate module, respectively.

$$\begin{aligned} V_{i,1} &\leftarrow \text{LLM}(\hat{q}_{i,1}, \mathcal{P}_i, I_v) \\ &\vdots \\ V_{i,|\hat{\mathcal{Q}}_i|} &\leftarrow \text{LLM}(\hat{q}_{i,|\hat{\mathcal{Q}}_i|}, \mathcal{P}_i, I_v), \end{aligned} \quad (7)$$

where each  $V_{i,j}$  consists of a binary label (i.e., Yes or No). For instance, if all  $V_{i,*}$  are determined "Yes" the grade is Useful. We present the full prompt of  $I_v$  in Table 9 in Appendix D. For  $\text{LLM}(\cdot)$ , we consider GPT-3.5 and GPT-4.

**Adaptive Generation** Once we get the verification results from Eq 7, if the  $\mathcal{P}_i$  is classified to Useful or PartialUseful, we decide to utilize the retrieved passages  $\mathcal{P}_i$  to generate a response by Eq 1 and 2. If  $\mathcal{P}_i$  is classified to Useless, we decide to only utilize the LLM’s internal knowledge to generate a response:  $\text{LLM}(q_i, I_1)$ . The full prompt of  $I_1$  is presented in Table 12 in Appendix D. This process enables the utilization of the most suitable approach tailored to each retrieval quality, which is beneficial to both accuracy and efficiency.

### 3.4 Discussion

**Efficiency.** We examine the factors contributing to DIVA’s strong efficiency in Figure 5, which illustrates the average number of input and output tokens per query when using the GPT-4 backbone. For a detailed explanation of the token consumption calculation process, please refer to Appendix A.4.2. **First, DIVA’s strong efficiency is largely due to the RD method.** Unlike Iterative RAG, which involves an average of 5.5 exploration steps per query and requires more than 12,000 tokens for input and 1,200 tokens for output, the RD method significantly reduces the number of tokens needed. **Second, although the RV method introduces some additional costs, these are acceptable compared to the complexity of Iterative RAG.** Moreover, RV enables the adaptive generation (AG) strategy, where the faster closed-book LLM is selectively used instead of RAG, further enhancing efficiency. As a result, DIVA, combin-

ing RD, RV, and AG, requires substantially less inference time and API costs.

**Technical Contribution.** Our paper is the first attempt to investigate how current RAG methods struggle when handling ambiguous queries, and introduces several novel methods specifically tailored to ambiguous queries: retrieval diversification (RD), and retrieval verification (RV). The primary innovation of RD lies in its unique pseudo-interpretation inference, which mimics human reasoning process, and retrieval method. Furthermore, our RV module offers a novel approach that formulates retrieval verification in the context of ambiguous queries. Therefore, our work provides novel insights and strategies targeted at the ambiguous questions, offering a robust and efficient solution to the issues in previous RAG approaches.

## 4 Experimental Setups

### 4.1 Datasets

Our proposed method and all baseline models are assessed using the ASQA (Stelmakh et al., 2022) and SituatedQA (Zhang and Choi, 2021) datasets. ASQA is a long-form QA dataset featuring ambiguous questions. SituatedQA is a short-form QA dataset featuring questions that specifically highlight ambiguities related to temporal and geographical contexts. We give these questions to the QA systems and assess how comprehensively the responses cover the provided possible interpretations of questions. Further details about the datasets are provided in the Appendix A.1.

### 4.2 Evaluation Metrics

**Metrics for QA.** Following Min et al. (2020), we mainly adopt F1-based metrics. For the short-form QA dataset (SituatedQA) we utilize F1 score. Given ASQA is the long-form QA dataset, following Stelmakh et al. (2022), we use Disambig-F1 (D-F1) score instead of F1. We further leverage ROUGE-L (R-L) to measure correctness of the long-form responses. Finally, Disambiguation-ROUGE (DR), combines R-L and D-F1 scores for overall performance.

**Metrics for Passage Retrieval.** Following Min et al. (2021), we use MRecall@ $k$  to evaluate the quality of retrieved passages.

For more details of the evaluation metrics, please refer to Appendix A.2.

### 4.3 Baselines

We compare our DIVA against relevant models, including fully-supervised LMs, few-shot closed

book LLMs, LLMs w/ RAG, and the adaptive generation. Specifically, fully-supervised LMs include the 1) **T5 closed-book** (Raffel et al., 2020), 2) **T5 w/ JPR** (Min et al., 2021), and 3) **PaLM** (Chowdhery et al., 2023) w/ **Soft Prompt Tuning**. Few-shot closed book LLMs include 4) **Vanilla Llama3**, **GPT-3.5-turbo**, and **GPT-4** and 5) **Query refinement** (Amplayo et al., 2022). Few-shot LLMs w/ RAG include 6) **Vanilla RAG** where we use RAC prompt in Kim et al. (2023), for 7) **Iterative RAG** we use the sota method ToC (Kim et al., 2023), for adaptive generation 8) **Self-RAG** (Asai et al., 2023), and for RAG with retrieval verification 9) **CRAG** (Yan et al., 2024). For more details of the baselines, please refer to Appendix A.3.

### 4.4 Implementation Details

In DIVA, the LLM is employed across three modules: retrieval diversification (Eqn 3), retrieval verification (Eqn 7), and adaptive response generation (Eqn 1, 2, and closed-book LLM). For adaptive response generation, we use the same LLM backbones as the other baselines. For the retrieval diversification and verification modules, we assess the performance of GPT-3.5 (gpt-35-turbo) and GPT-4 (gpt-4-0613) across them, ultimately opting to use GPT-4 for both modules in the ASQA dataset and GPT-3.5 for both modules in the SituatedQA dataset in all experiments. However, as demonstrated in Section 5.4, other LLMs also perform effectively in these modules. For other implementation details, please refer to Appendix A.4.

## 5 Experimental Results and Analyses

### 5.1 Main Results

In this section, we assess the effectiveness of DIVA on ambiguous and unambiguous questions.

Table 1 presents the long-form ambiguous QA performance of baselines and DIVA on the development set of ASQA.

**First, DIVA outperforms the sota baseline, Iterative RAG, in terms of both accuracy and efficiency of response generation.** Our method enhances Vanilla RAG framework by incorporating retrieval diversification and adaptive generation strategies that address low-quality retrieval and improve performance. It is also more efficient, requiring significantly less computational overhead and achieving 1.5x - 3x greater efficiency in inference time across various LLM backbones compared to Iterative RAG. Overall, our method produces more accurate and diverse interpretations without the cumbersome iterative exploration process.

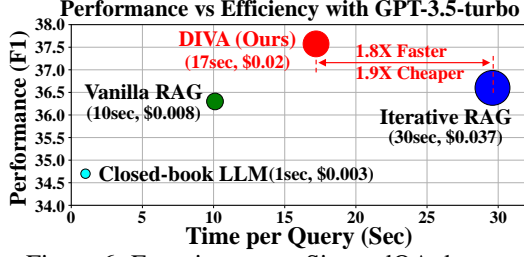


Figure 6: Experiments on SituatedQA dataset.

**Second, DIVA outperforms the recent retrieval verifying method, CRAG, across all metrics, including inference time.** CRAG underperforms even compared to Vanilla RAG, despite its verification and correction mechanisms. This suggests that CRAG’s verification and correction methods are not well-suited to handling ambiguous queries, resulting in degraded passage retrieval performance. These findings emphasize the need for a RAG method specifically designed for ambiguous queries, demonstrating the practicality and effectiveness of DIVA in such scenarios.

**Third, DIVA demonstrates good adaptability in switching out the underlying LLM backbones.** DIVA consistently enhances Vanilla RAG with its RD and AG modules across different LLM backbones regardless of their model sizes (Llama3-8B to GPT-4), demonstrating its adaptability and wide applicability. This suggests that DIVA can easily integrate with more advanced LLMs in the future.

Fig 6 shows the performance and efficiency of baselines and DIVA on the SituatedQA test set for short-form ambiguous QA tasks. All experimental results align with those seen in Table 1, demonstrating strong generalizability of DIVA across different types of ambiguous questions.

**Finally, DIVA exhibits strong performance on unambiguous questions as well, i.e., NQ dataset (Kwiatkowski et al., 2019), highlighting its broad applicability.** For detailed results and explanations, please refer to Appendix C.2.

## 5.2 Ablation Studies

To evaluate the importance of each component of DIVA, namely retrieval diversification (RD) and adaptive generation (AG), we incrementally add them to Vanilla RAG (row 2 in Table 2). Table 2 reveals the following insights: **1)** RAG (row 2) with the closed-book LLM (row 1) significantly enhances the ability to handle ambiguity in questions. **2)** Implementing the RD module (row 3) enhances all performance metrics, demonstrating that RD effectively diversifies and improves the quality of retrieved passages, thereby enhancing the RAG framework. **3)** Incorporating the AG mod-

	R-L	D-F1	DR	Time
<b>Fully-Supervised</b>				
T5-Large Closed-Book*	33.5	7.4	15.7	-
T5-Large w/ JPR*	43.0	26.4	33.7	-
PaLM w/ Soft Prompt Tuning**	37.4	27.8	32.1	-
<b>Few-shot Prompting: Closed-Book LLM</b>				
Llama3-8B-Instruct	31.1	25.6	28.2	-
Llama3-70B-Instruct	35.7	36.4	36.0	30.5
GPT-3.5-turbo	38.8	34.0	36.3	2.0
+ Query Refinement	37.5	34.8	36.1	5.2
GPT-4	39.0	38.5	38.7	5.9
+ Query Refinement	39.6	39.3	39.4	10.0
<b>Few-shot Prompting: LLM w/ RAG</b>				
Self-RAG-13B	35.4	26.0	30.4	4.1
CRAG (GPT-4)	40.1	39.6	39.9	34.4
<b>Llama3-8B-Instruct</b>				
— Vanilla RAG	38.2	35.4	36.8	-
— Iterative RAG (ToC)	37	36.3	36.6	-
— DIVA (Ours)	<b>38.9</b>	<b>35.7</b>	<b>37.3</b>	-
<b>Llama3-70B-Instruct</b>				
— Vanilla RAG	40.2	40.0	40.1	42.3
— Iterative RAG (ToC)	39.5	40.4	39.9	140.5
— DIVA (Ours)	<b>40.4</b>	<b>41.4</b>	<b>40.9</b>	<b>50.6</b>
<b>GPT-3.5-turbo</b>				
— Vanilla RAG	41.2	37.5	39.3	11.2
— Iterative RAG (ToC)	40.1	38.5	39.3	31.5
— DIVA (Ours)	<b>42.1</b>	<b>38.9</b>	<b>40.5</b>	<b>19.8</b>
<b>GPT-4</b>				
— Vanilla RAG	41.5	39.6	40.6	18.9
— Iterative RAG (ToC)	38.5	41.8	40.1	53.1
— DIVA (Ours)	<b>42.4</b>	<b>42.0</b>	<b>42.2</b>	<b>27.1</b>

\* results from Stelmakh et al. (2022)

\*\* results from Amplayo et al. (2022)

Table 1: Experiments on ASQA dataset. Baselines are either fully-supervised or 5-shot prompted. The metric Time indicates inference time (sec) per query. We emphasize our results in bold, for easy comparisons.

ule (row 4) also boosts all metrics, showing that the retrieval verification method accurately identifies Useless passages. Additionally, this supports our finding in Sec 2 that when retrieved passages are of extremely low quality, the internal knowledge of LLMs proves more advantageous than RAG.

## 5.3 Retrieval Analysis

We evaluate the effectiveness of our proposed RD method in Table 3 using MRecall@ $k$  (Min et al., 2021). Vanilla RAG (row 1) involves basic retrieval of passages using a given question  $q_i$ . "+ RD" (row 3) applies the RD method to row 1, using *pseudo-interpretations* generated by our proposed instructions (i.e.,  $I_p$  and  $I_a$ ). Row 2 uses the RD method with *pseudo-interpretations* generated by the LLM query rewriter as described in Ma et al. (2023) using simple instructions. "+ Oracle" (row 4) applies RD to Vanilla RAG using ground-truth

Row	Component			GPT-3.5-turbo			GPT-4		
	RAG	RD	AG	R-L	D-F1	DR	R-L	D-F1	DR
1	✗	✗	✗	38.8	34.0	36.3	39.0	38.5	38.7
2	✓	✗	✗	41.2	37.5	39.3	41.5	39.6	40.6
3	✓	✓	✗	42.1	38.5	40.2	42.3	41.0	41.7
4	✓	✓	✓	42.1	38.9	40.5	42.4	42.0	42.2

Table 2: Ablation studies on ASQA dataset.

Row	Method	MRecall@ $k=5$	D-F1	
			GPT-3.5-turbo	GPT-4
1	Vanilla RAG	35.2	37.5	39.6
2	+ Ma et al. (2023)	36.1	37.0	40.4
3	+ RD (Ours)	<b>37.0</b>	<b>38.5</b>	<b>41.0</b>
4	+ Oracle	41.5	-	-

Table 3: Retrieval accuracy and corresponding QA performance on ASQA dataset.

interpretations from the ASQA dataset.

We observe that **1)** adding RD leads to significant improvements of MRecall and D-F1 score compared to Vanilla RAG, demonstrating RD effectively addresses low-quality retrieval issue and then improve the QA performance. **2)** "+ RD" outperforms "+ Ma et al. (2023)" showing the superiority of our carefully designed instruction in inferring *pseudo-interpretations*. **3)** "+ Oracle" (row 4) significantly outperforms RD, indicating that when more advanced LLMs are available in the future there is potential for RD to improve in accurately inferring *pseudo-interpretations*.

#### 5.4 Sensitivity Analysis

For the retrieval diversification (RD) and retrieval verification (RV) modules, we explore how their performance is affected by the choice of LLM. We evaluate the impact of using GPT-3.5 and GPT-4 across both modules, comparing the overall QA performance against the sota baseline, ToC (Kim et al., 2023), on the ASQA and SituatedQA datasets. Fig 7(a) and (b) represent using GPT-3.5 and GPT-4 as the response generation models on the ASQA dataset, respectively. Fig 7(c) represents using GPT-3.5 as the response generation model on the SituatedQA dataset.

In Fig 7, we observe: **1)** DIVA consistently outperforms ToC, regardless of the LLM model used in each module. **2)** While the RD module shows very stable results, the RV module appears relatively sensitive to the choice of LLM. This highlights that verifying the quality of retrieved passages for ambiguous questions requires more powerful natural language understanding ability, underscoring the need for future work to alleviate the dependency on the choice of LLM. Based on these results, we argue that DIVA is a general framework

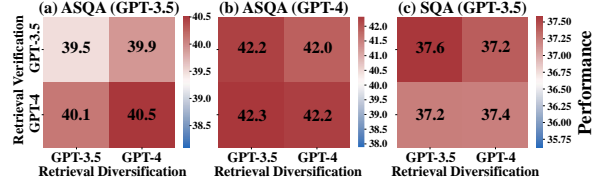


Figure 7: Sensitivity analysis of LLM backbone model in retrieval diversification and verification modules. Red-white-blue means outperformance, on-par, and underperformance compared with Iterative RAG (ToC) in terms of DR for ASQA and F1 for SituatedQA.

that is robust across different LLMs.

#### 5.5 Case Study

We conduct a case study to qualitatively compare the reasoning chains of Iterative RAG, ToC (Kim et al., 2023), and DIVA. Figure 8 illustrates the reasoning chains of Iterative RAG, ToC (Kim et al., 2023), and DIVA using the ASQA question, "The movement of food in the food pipe is called?". In panel (a), the answer "Peristalsis" is easily covered during the first exploration, whereas "Swallowing" requires six steps of passage retrieval and LLM reasoning for exploration. In contrast, panel (b) shows that our *pseudo-interpretations* include both interpretations, with the RD retrieving passages that encompass all necessary information. Consequently, the LLM efficiently extracts all plausible interpretations from the retrieved passages without the need for the cumbersome iterative exploration process.

Moreover, we analyze failure cases to provide valuable insights into the limitations and potential improvements of DIVA by identifying instances where it underperforms. Detailed results and explanations are presented in Appendix C.1.

### 6 Related Work

**RAG for Ambiguous Question.** To tackle the ambiguity inherent in certain questions, earlier studies (Min et al., 2021; Gao et al., 2020; Shao and Huang, 2021; Sun et al., 2023) necessitated the fine-tuning of models using extensive training datasets. Recently, some studies have leveraged LLM to generate comprehensive responses through an in-context learning. For example, RAC (Kim et al., 2023) instructs LLM to extract plausible interpretations and answers from provided passages. However, they overlook the problem of low-quality retrieval, which results in significant performance drops. To tackle this issue, ToC (Kim et al., 2023) explores missing interpretations by an iterative passage retrieval and LLM reasoning. However, the iterative process incurs significant computational overhead. **Query Reformulation.** Our pseudo-interpretation



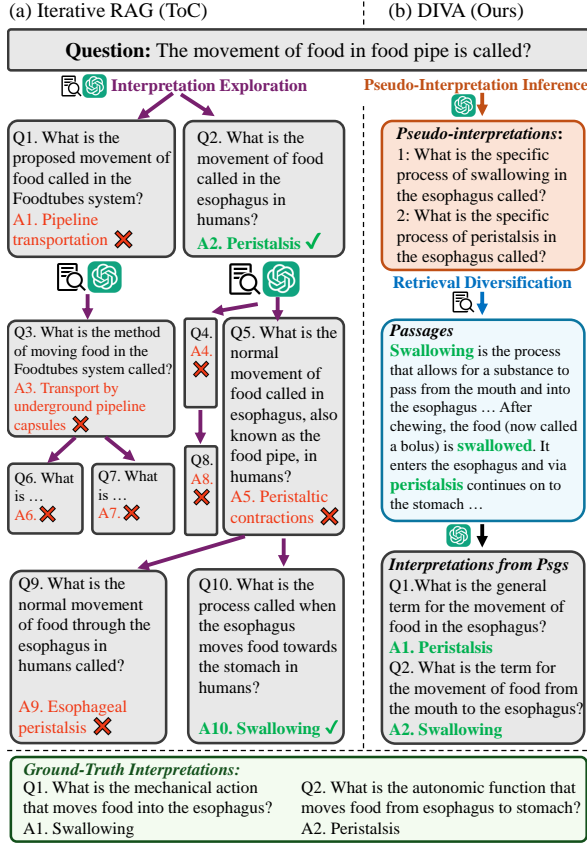


Figure 8: Case study with GPT-4.

inference may appear to share similarities with existing query reformulation approaches. The first line of research is query decomposition (Min et al., 2019; Khot et al., 2022), which breaks down complex queries containing multi-hop relations or an overabundance of information. In contrast, our work addresses the opposite challenge, ambiguity, caused by a lack of information within the query, rather than the overabundance of information. Therefore, query decomposition and pseudo-interpretation inference are fundamentally different in purpose. Another line is query rewriting. Ma et al. (2023) utilize an LLM and naively designed prompt to rewrite the given query to improve the quality of retrieval. Compared to Ma et al. (2023), the primary innovation of DIVA lies in its unique and well-designed prompting method, which imitates human reasoning chains to infer the pseudo-interpretations from ambiguous queries.

**Retrieval Quality Verification.** Many studies have noted that low-quality retrieval introduces significant irrelevant information to the RAG framework and have proposed various solutions. Self-RAG (Asai et al., 2023) fine-tunes LLM to generate a reflection token that assesses the relevance of a passage to the question at hand. Llatrival (Li et al., 2023) employs LLM to check if retrieved

passages sufficiently support the answer, updating them if they are of low quality. Meanwhile, CRAG (Yan et al., 2024) trains a lightweight verifier to evaluate the quality of retrieved passages, making corrections if they fall below a set threshold. Please note that our work focuses on retrieval verification, distinguishing it from other methods such as CoVe (Dhuliawala et al., 2023) and Verify-and-Edit (Zhao et al., 2023), which do not target this aspect. For a more detailed discussion, please refer to Appendix B.2.

**Adaptive Generation.** Numerous studies have examined adaptive strategies that dynamically determine the need for retrieval, utilizing only the internal knowledge of LLMs when unnecessary (Mallen et al., 2023; Feng et al., 2023). Mallen et al. (2023) used an empirical method to retrieval, activating relying on the frequency of entity. AdaptiveRAG (Jeong et al., 2024) dynamically chooses the optimal response generation strategy tailored to the complexity of the query. TA-ARE (Zhang et al., 2024) uses in-context learning to assess whether a query necessitates retrieval.

Compared with recent studies that either overlook or inefficiently address the issue of low-quality retrieval in ambiguous questions, we introduce the retrieval diversification method efficiently retrieves higher quality passages without relying on cumbersome iterative processes. Additionally, we propose retrieval verification and adaptive generation strategies specifically designed for ambiguous questions, while the existing works overlook these important challenge of ambiguous questions. To the best of our knowledge, this paper is the first effort to thoroughly analyze and address the problem of low-quality retrieval in the context of ambiguous questions and its potential solutions.

## 7 Conclusion

In this study, we examined the shortcomings of the current RAG-based method in dealing with ambiguous questions, specifically its low-quality retrieval and inefficiency. Our proposed framework, DIVA, effectively diversifies the retrieved passages to capture various interpretations, verifies their quality, and adapts the most appropriate approach based on that quality. This strategy improves QA performance while minimizing inefficiency.

## Limitations

While DIVA demonstrates clear advantages in effectiveness and efficiency through retrieval diversification and adaptive generation, its design is specif-

ically tailored for ambiguous questions. In real-world QA systems, where queries can be a mix of ambiguous and unambiguous, the applicability of DIVA may be limited. However, recent work has introduced methods to classify whether a query is ambiguous (Cole et al., 2023), which leads to utilizing the suitable approach according to its ambiguity. Although Cole et al. (2023) proposed simple approaches, there is still significant potential to enhance these methods using advanced techniques like in-context learning and RAG. Future research could focus on developing systematic approaches for classifying the ambiguity of queries. Furthermore, the performance of our proposed retrieval verification module is somewhat sensitive to the choice of LLM. Specifically, it tends to work better with GPT-4 than with GPT-3.5, though this may negatively impact the efficiency of DIVA. Therefore, future work should focus on developing a more efficient and robust retrieval quality verifier LLM, tailored to handling ambiguous questions, to enhance both effectiveness and efficiency.

## Ethics Statement

Given that DIVA is built on the RAG framework of QA systems, it is important to consider the following points: (1) the retrieved passages may contain offensive or harmful content, which could result in similarly harmful responses, and (2) user queries themselves may be offensive or harmful. Therefore, developing methods to detect harmful user queries and selectively retrieve passages that are free from harmful content could be a crucial focus for future research.

## Acknowledgements

This work was supported by Institute of Information & Communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (RS-2023-00216011, Development of artificial complex intelligence for conceptually understanding and inferring like human) and National Research Foundation of Korea(NRF) funded by Ministry of Science and ICT (NRF-2022M3J6A1063021).

## References

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

Reinald Kim Amplayo, Kellie Webster, Michael Collins, Dipanjan Das, and Shashi Narayan. 2022. Query refinement prompts for closed-book long-form question answering. [arXiv preprint arXiv:2210.17525](#).

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

Tom B Brown. 2020. Language models are few-shot learners. [arXiv preprint ArXiv:2005.14165](#).

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

Jeremy R Cole, Michael JQ Zhang, Daniel Gillick, Julian Martin Eisenschlos, Bhuwan Dhingra, and Jacob Eisenstein. 2023. Selectively answering ambiguous questions. [arXiv preprint arXiv:2305.14613](#).

Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. 2023. Chain-of-verification reduces hallucination in large language models. [arXiv preprint arXiv:2309.11495](#).

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. [arXiv preprint arXiv:2407.21783](#).

Zhangyin Feng, Weitao Ma, Weijiang Yu, Lei Huang, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. 2023. Trends in integration of knowledge and large language models: A survey and taxonomy of methods, benchmarks, and applications. [arXiv preprint arXiv:2311.05876](#).

Yifan Gao, Henghui Zhu, Patrick Ng, Cicero Nogueira dos Santos, Zhiguo Wang, Feng Nan, De-jiao Zhang, Ramesh Nallapati, Andrew O Arnold, and Bing Xiang. 2020. Answering ambiguous questions through generative evidence fusion and round-trip prediction. [arXiv preprint arXiv:2011.13137](#).

Gautier Izacard and Edouard Grave. 2021. [Leveraging passage retrieval with generative models for open domain question answering](#). In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 874–880, Online. Association for Computational Linguistics.

Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong C Park. 2024. Adaptive-rag: Learning to adapt retrieval-augmented large language models through question complexity. [arXiv preprint arXiv:2403.14403](#).

- Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. [TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.
- 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*.
- Omar Khattab and Matei Zaharia. 2020. [Colbert: Efficient and effective passage search via contextualized late interaction over bert](#).
- Tushar Khot, Harsh Trivedi, Matthew Finlayson, Yao Fu, Kyle Richardson, Peter Clark, and Ashish Sabharwal. 2022. [Decomposed prompting: A modular approach for solving complex tasks](#). *arXiv preprint arXiv:2210.02406*.
- Gangwoo Kim, Sungdong Kim, Byeongguk Jeon, Joon-suk Park, and Jaewoo Kang. 2023. [Tree of clarifications: Answering ambiguous questions with retrieval-augmented large language models](#). *arXiv preprint arXiv:2310.14696*.
- Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. [Natural questions: A benchmark for question answering research](#). *Transactions of the Association for Computational Linguistics*, 7:452–466.
- Angeliki Lazaridou, Elena Gribovskaya, Wojciech Stokowiec, and Nikolai Grigorev. 2022. [Internet-augmented language models through few-shot prompting for open-domain question answering](#). *arXiv preprint arXiv:2203.05115*.
- Xiaonan Li, Changtai Zhu, Linyang Li, Zhangyue Yin, Tianxiang Sun, and Xipeng Qiu. 2023. [Llatrieval: Llm-verified retrieval for verifiable generation](#). *arXiv preprint arXiv:2311.07838*.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. [Roberta: A robustly optimized bert pretraining approach](#). *arXiv preprint arXiv:1907.11692*.
- Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. 2023. [Query rewriting for retrieval-augmented large language models](#). *arXiv preprint arXiv:2305.14283*.
- Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. [When not to trust language models: Investigating effectiveness of parametric and non-parametric memories](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9802–9822, Toronto, Canada. Association for Computational Linguistics.
- Sewon Min, Kenton Lee, Ming-Wei Chang, Kristina Toutanova, and Hannaneh Hajishirzi. 2021. [Joint passage ranking for diverse multi-answer retrieval](#). *arXiv preprint arXiv:2104.08445*.
- Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2020. [Ambigqa: Answering ambiguous open-domain questions](#). *arXiv preprint arXiv:2004.10645*.
- Sewon Min, Victor Zhong, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2019. [Multi-hop reading comprehension through question decomposition and rescoring](#). *arXiv preprint arXiv:1906.02916*.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. [Exploring the limits of transfer learning with a unified text-to-text transformer](#). *Journal of machine learning research*, 21(140):1–67.
- Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlga, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. [In-context retrieval-augmented language models](#). *Transactions of the Association for Computational Linguistics*, 11:1316–1331.
- Nils Reimers and Iryna Gurevych. 2019. [Sentence-bert: Sentence embeddings using siamese bert-networks](#). *arXiv preprint arXiv:1908.10084*.
- Zhihong Shao and Minlie Huang. 2021. [Answering open-domain multi-answer questions via a recall-then-verify framework](#). *arXiv preprint arXiv:2110.08544*.
- Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. 2023. [Replug: Retrieval-augmented black-box language models](#). *arXiv preprint arXiv:2301.12652*.
- Ivan Stelmakh, Yi Luan, Bhuwan Dhingra, and Ming-Wei Chang. 2022. [Asqa: Factoid questions meet long-form answers](#). *arXiv preprint arXiv:2204.06092*.
- Weiwei Sun, Hengyi Cai, Hongshen Chen, Pengjie Ren, Zhumin Chen, Maarten de Rijke, and Zhaochun Ren. 2023. [Answering ambiguous questions via iterative prompting](#). *arXiv preprint arXiv:2307.03897*.
- Shi-Qi Yan, Jia-Chen Gu, Yun Zhu, and Zhen-Hua Ling. 2024. [Corrective retrieval augmented generation](#). *arXiv preprint arXiv:2401.15884*.

Michael Zhang and Eunsol Choi. 2021. [SituatingQA: Incorporating extra-linguistic contexts into QA](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7371–7387, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Zihan Zhang, Meng Fang, and Ling Chen. 2024. Retrievalqa: Assessing adaptive retrieval-augmented generation for short-form open-domain question answering. [arXiv preprint arXiv:2402.16457](#).

Ruochen Zhao, Xingxuan Li, Shafiq Joty, Chengwei Qin, and Lidong Bing. 2023. Verify-and-edit: A knowledge-enhanced chain-of-thought framework. [arXiv preprint arXiv:2305.03268](#).

---

**Algorithm 1** diversify-verify-adapt (DIVA)

---

**Input:** Question  $q_i$ , large language model  $\text{LLM}(\cdot)$ , the retriever  $\mathcal{R}(\cdot)$ , candidate passages  $\mathcal{C}_i$

**Output:** generated response  $r_i$

- 1:  $\hat{\mathcal{Q}}_i \leftarrow$  Use  $\text{LLM}(\cdot)$  function to infer the *pseudo-interpretations* by Eq.3
  - 2:  $\tilde{\mathcal{P}}_i \leftarrow$  Use  $\mathcal{R}(\cdot)$  function to retrieve relevant and diverse passages by Eq.4
  - 3:  $\mathcal{S}_i \leftarrow \{\}$
  - 4: **for**  $j = 1$  to  $|\tilde{\mathcal{P}}_i|$  **do**
  - 5:    $S(\tilde{p}_{i,j}) \leftarrow$  Obtain noise score by Eq.5
  - 6:    $\mathcal{S}_i \leftarrow \mathcal{S}_i \cup S(\tilde{p}_{i,j})$
  - 7: **end for**
  - 8:  $\mathcal{P}_i \leftarrow$  Select top- $K$  passages from  $\tilde{\mathcal{P}}_i$  based on the score  $\mathcal{S}_i$
  - 9:  $\mathcal{V}_i \leftarrow \{\}$
  - 10: **for**  $j = 1$  to  $|\hat{\mathcal{Q}}_i|$  **do**
  - 11:    $V_{i,j} \leftarrow$  Use  $\text{LLM}(\cdot)$  function to verify  $\mathcal{P}_i$  to  $\hat{q}_{i,j}$  by Eq.7
  - 12:    $\mathcal{V}_i \leftarrow \mathcal{V}_i \cup V_{i,j}$
  - 13: **end for**
  - 14: **if**  $\mathcal{V}_i$  is Useful or PartialUseful **then**
  - 15:    $r_i \leftarrow$  Generate response using  $\mathcal{P}_i$  by Eq.1 and 2
  - 16: **else**
  - 17:    $r_i \leftarrow \text{LLM}(q_i, I_1)$
  - 18: **end if**
  - 19: **return**  $r_i$
- 

## A Experimental details

### A.1 Datasets

Our proposed method and all baseline models are assessed using the ASQA (Stelmakh et al., 2022) and SituatingQA (Zhang and Choi, 2021) datasets. ASQA is a long-form QA dataset derived from a subset of ambiguous questions in the AmbigNQ dataset (Min et al., 2020). The ASQA dataset contains 6,316 ambiguous questions and their corresponding comprehensive long-form answers that contain all plausible answers, split into 4,353 for training, 948 for development, and 1,015 for testing. SituatingQA is a short-form QA dataset featuring questions that specifically highlight ambiguities related to temporal and geographical contexts. In this dataset, each question is subject to multiple interpretations, with corresponding answers varying by context. We give these questions to the QA systems and assess how comprehensively the responses cover the possible interpretations.



## A.2 Evaluation Metrics

**Metrics for QA.** For both datasets, following previous studies on ambiguous QA (Min et al., 2020), we mainly adopt F1-based metric. Specifically, for the short-form QA dataset (SituatQA) we measure F1 based on the precision and recall between the ground-truth answers and the generated responses. Given ASQA is the long-form QA dataset, following Stelmakh et al. (2022), we use Disambig-F1 (D-F1), which assesses the factual accuracy of long-form responses, instead of F1. Using a RoBERTa model (Liu et al., 2019) trained on SQuAD2.0, we extract short answers from the generated long-form responses and compare them to the ground-truth disambiguation questions (DQs). The F1 score of these extracted answers indicates whether the long-form answers contain correct information. We further leverage ROUGE-L (R-L) to measure correctness of the generated long-form responses to the ground-truth long-form answers. Finally, Disambiguation-ROUGE (DR), combines R-L and D-F1 scores as a geometric mean for overall performance.

**Metrics for Passage Retrieval.** Following Min et al. (2021), we use MRecall@ $k$  to evaluate the quality of retrieved passages by considering retrieval to be successful if all answers or at least  $k$  answers in the plausible answer set are recovered by the retrieved passages.

## A.3 Baselines

For all baselines and DIVA, due to the significant costs associated with evaluating RAG models, we perform experiments with a single run.

We describe the details of models as follows:

- 1) **T5 closed-book.** Stelmakh et al. (2022) fine-tuned T5-large (Raffel et al., 2020) to generate long-form response on the whole train set.
- 2) **T5 w/ JPR.** Stelmakh et al. (2022) fine-tuned T5-large (Raffel et al., 2020) with JPR (Min et al., 2021), fully trained dense retriever for ambiguous QA, to generate long-form response on the whole train set.
- 3) **PaLM w/ Soft Prompt Tuning.** Amplayo et al. (2022) employed a prompt engineering method to PaLM (Chowdhery et al., 2023) that learn the soft prompts in the closed-book setup.
- 4) **Closed-book LLM.** Closed-book LLM indicates the traditional few-shot prompting method used in Brown (2020). We consider the backbone LLM as Llama3-70B-Instruct, GPT-3.5, and GPT-4.
- 5) **Query refinement.** Inspired by Amplayo

et al. (2022), we developed an in-context learning method within a closed-book setup. First, we prompt the LLM to refine ambiguous questions into multiple possible interpretations. These interpretations are then used as in-context examples for the LLM to generate a response that addresses all potential interpretations. We consider the backbone LLM as Llama3-70B-Instruct, GPT-3.5, and GPT-4.

6) **Vanilla RAG.** In this method, we begin by retrieving the top 5 relevant passages based on the frozen SentenceBERT similarity between the given query and candidate passages from Wikipedia. We then use the RAC prompt from Kim et al. (2023) to extract interpretations and generate corresponding answers. We consider the backbone LLM as Llama3-70B-Instruct, GPT-3.5, and GPT-4.

7) **Iterative RAG.** For this approach, we employ the state-of-the-art method ToC (Kim et al., 2023) for handling ambiguous QA. Specifically, ToC iteratively constructs a tree of possible interpretations for the ambiguous question using few-shot prompting that leverages external knowledge, and then uses this tree to generate a long-form response. Following the authors’ implementation, we set the tree’s maximum depth to 3 and the maximum number of nodes to 10. It is important to note that we do not use the tree pruning method in our implementation, as we observe that adding this method notably degrades the QA performance. The retrieval settings are identical to those used in Vanilla RAG. We consider the backbone LLM as Llama3-8B-Instruct, Llama3-70B-Instruct (Dubey et al., 2024), GPT-3.5, and GPT-4.

8) **Self-RAG.** The LLM is trained to adaptively manage retrieval and generation, initiating retrieval when a special token is predicted above a certain threshold, followed by generating the answer. We consider the model trained on Llama2-13B.

9) **CRAG.** While the original implementation of CRAG utilized Llama-2-7B, we use GPT-4 as the backbone LLM for a fair comparison, ensuring consistency with the DIVA setup. In the CRAG implementation, we first retrieve relevant passages for a given query, following the same procedure as Vanilla RAG. Next, we apply CRAG’s retrieval verification and correction procedures to refine these passages. The corrected passages are then fed into GPT-4 using the same instructions as in the Vanilla RAG framework to generate the final response to the query.

#### A.4 Implementations Details

Since DIVA utilizes few-shot prompting, we dynamically select  $k$ -shot examples through nearest neighbor search and incorporate them into the prompt, following the approach in Kim et al. (2023) using dsp package (Khattab et al., 2022). For the retrieved passages  $\mathcal{P}_i$ , we set the number of passages  $|\mathcal{P}_i|$  to 5. We use GPT-4 for both the retrieval diversification and verification steps. For adaptive response generation, we use the same LLM backbones as the other baselines. For the retrieval diversification and verification modules, we assess the performance of GPT-3.5 (gpt-35-turbo) and GPT-4 (gpt-4) across them, ultimately opting to use GPT-4 for both modules in the ASQA dataset and GPT-3.5 for both modules in the SituatedQA dataset for all experiments. The APIs provided by Microsoft Azure<sup>3</sup> are employed for GPT-3.5-turbo and GPT-4, with the following settings: max tokens set to 300, top- $p$  to 1.0, and temperature to 0.3.

##### A.4.1 Retrieval Process

To retrieve relevant passages for the given question, we follow the method utilized by Kim et al. (2023). Specifically, we first gather relevant Wikipedia documents for the question using two retrieval systems: ColBERT (Khattab and Zaharia, 2020) and the Bing search engine<sup>4</sup>. After compiling a set of passages, we rerank and select the top- $k$  passages. For reranking, we utilize SentenceBERT (Reimers and Gurevych, 2019), pre-trained on MS-Marco, as the backbone.

##### A.4.2 Token Consumption Calculation

In this subsection, we explain the overall process of each method and the token consumption calculation on GPT-4. Notably, the token count per query was determined by averaging the input and output tokens across all test queries.

**Vanilla RAG** is formally described as a sequence of LLM functions as shown in Eq 1 and 2. In Eq 1, given the relevant  $\mathcal{P}_i$ , the LLM is first prompted with the question  $q_i$  and  $\mathcal{P}_i$  to extract all plausible interpretations  $\mathcal{Q}_i$  and their corresponding answers  $\mathcal{A}_i$ . During this function call, the average number of input tokens is 1,902, while the average number of output tokens is 177. Note that the input text include task description, few-shot demos, retrieved passages, and given questions, leading to substantial token consumption. Next, in Eq 2, based on the  $\mathcal{Q}_i$  and  $\mathcal{A}_i$ , the LLM is prompted

to consolidate them with  $q_i$  and  $\mathcal{P}_i$  to generate a response  $r_i$ . In this function call, the averaged input tokens are 1,963, and the output tokens are 249. **Consequently, for Vanilla RAG, the total token consumption is 3,865 input tokens and 426 output tokens.**

**Iterative RAG** is formally represented as iterative LLM function calls in Eq 1, followed by a single LLM call in Eq 2. Specifically, after obtaining multiple plausible interpretations  $\mathcal{Q}_i$  and their answers  $\mathcal{A}_i$  from Eq 1, each interpretation in  $\mathcal{Q}_i$  is used for an additional LLM call in Eq 1. This process is repeated, constructing a tree-like structure, until the stopping criterion is met. On average, during this iterative process, the total number of LLM calls converges to 5.5. As a result, the average number of input and output tokens during this process are 10,627 and 936, respectively. Next, same as Vanilla RAG in Eq 2, based on all  $\mathcal{Q}_i$  and  $\mathcal{A}_i$  collected from the iterative process, the LLM is prompted to consolidate them with  $q_i$  and  $\mathcal{P}_i$  to generate a response  $r_i$ . In this LLM call, the average number of input tokens is 2,187, and the output tokens is 271. **Consequently, for Iterative RAG, the total token consumption is 12,814 input tokens and 1,207 output tokens.**

For **DIVA**, in addition to the calls required by Vanilla RAG, additional LLM calls are introduced through the operations of the RD (Retrieval Diversification) and RV (Retrieval Verification) modules. These additional operations add to the overall token usage. More specifically, in RD, we require two LLM calls to infer a set of multiple pseudo-interpretations, where the number of input and output tokens are 2,440 and 117, respectively. Following this, the retrieval of relevant passages based on the inferred pseudo-interpretations does not require any LLM calls. Verifying the set of retrieved passages requires the same number of LLM calls as the number of pseudo-interpretations. It is important to note that the retrieved passages are concatenated into a single passage before the verification step, allowing for an efficient LLM call process for each pseudo-interpretation. For the verification step, the number of input and output tokens are 1,878 and 3, respectively. It is important to note that if the verifier determines the retrieved passages are not useful, the response is generated using a Closed-book LLM instead of Vanilla RAG. The token consumption for the Closed-book LLM is significantly lower, with 913 input tokens and 123 output tokens. **Consequently, for DIVA, the total token consumption is 7,873 input tokens and 515**

<sup>3</sup><https://azure.microsoft.com/>

<sup>4</sup><https://www.microsoft.com/bing>

output tokens.

Method	# Input Tokens	# Output Tokens
Closed-book LLM	913	123
Vanilla RAG	3,865	426
Iterative RAG	12,814	1,207
DIVA	7,873	515

Table 4: The number of input and output token consumption of each method.

## B Additional Discussion

### B.1 Regarding the Pseudo-interpretation Inference

We initially experimented with combining the ambiguity detection and pseudo-interpretation inference into a single step to simplify the process. Specifically, we provided GPT-4 with newly designed instructions to execute both steps simultaneously with appropriate few-shot demonstrations for inferring pseudo-interpretations. However, this approach performed significantly worse than our current method of separating the steps. The primary reason for this performance drop is that handling both tasks simultaneously imposes a substantial burden on a single LLM. Since each step requires detailed task descriptions and specific few-shot demonstrations, merging them results in an overload of information that negatively affects the model’s reasoning process. This also highlights the challenges in effectively exploring various interpretations without an iterative approach (i.e., iterative RAG). Despite this, our proposed Retrieval Diversification (RD) method still efficiently infers pseudo-interpretations within the two-step framework and achieves strong results without the need for iterative processes. This emphasizes the effectiveness and efficiency of our reasoning chain design, even when tasks are separated for clarity and precision.

### B.2 Comparison to Verification Methods

#### B.2.1 Comparison to Chain-of-Verification

The verification modules in DIVA and Chain-of-Verification (CoVe) (Dhuliawala et al., 2023) differ significantly in both their purpose ("why to use"), target ("where to use"), and timing ("when to use").

**Purpose ("why to use"):** The CoVe approach focuses on assessing the correctness of a generated response, ensuring the final output is accurate. In contrast, DIVA’s verification module is designed to assess the relevance of retrieved passages within the RAG framework before any response is generated. These methods are therefore tailored for

entirely different objectives—CoVe targets post-response accuracy, whereas DIVA emphasizes pre-response relevance.

**Target ("where to use"):** The verifier in CoVe operates on the generated responses, whereas DIVA’s verifier focuses on the retrieved passages. Given that these targets possess distinct characteristics and objectives, each verifier is uniquely designed to effectively capture the verification rationale relevant to its specific target. This fundamental difference between CoVe and DIVA distinguishes the two approaches, making it challenging for the verifiers to be compatible or interchangeable.

**Timing ("when to use"):** CoVe performs verification after the response has been generated and presented to the user. In contrast, DIVA operates earlier in the pipeline by verifying the retrieved passages before generating a response. Therefore, DIVA is cost-efficient as it can anticipate whether a generated response is likely to be incorrect before the response is even produced. This allows DIVA to avoid unnecessary response generation and associated costs, enabling the application of an optimal response strategy for the situation.

Additionally, DIVA introduces a novel aspect in its verification, specifically designed for handling ambiguous queries. It employs pseudo-interpretations to evaluate how well the retrieved passages encompass multiple interpretations of the question. This approach is distinct from CoVe and further enhances the novelty of DIVA.

#### B.2.2 Comparison to Verify-and-Edit

There are significant differences in the purposes of the verification modules within DIVA and Verify-and-Edit (Zhao et al., 2023). The Verify-and-Edit framework aims to assess the correctness of a generated chain of thought (CoT) and edit the CoT using retrieved external knowledge, ensuring an accurate reasoning process. On the other hand, DIVA’s verification module is tailored to evaluate the relevance of retrieved passages within the RAG framework.

These distinct goals highlight that the two methods are not only conceptually different but also serve different objectives. Therefore, our proposed verification module could be integrated into the Verify-and-Edit pipeline to improve the robustness of its editing process. Specifically, since the effectiveness of the Verify-and-Edit framework heavily relies on external knowledge for accurate editing, ensuring the relevance of retrieved passages is crucial. When there is ambiguity in the premise of the

CoT, DIVA’s verification module can verify the retrieved passages’ relevance, enhancing the overall editing process.

## C Additional Experiments

### C.1 Failure Cases

We examined 50 randomly selected samples in ASQA from those with a D-F1 value below 0.5, where D-F1 ranges from 0 to 1.

The first failure scenario occurs when the retrieval diversification (RD) module underperforms, accounting for 22 out of the 50 samples. These failures arise from errors in the generated pseudo-interpretations or the inherent limitations of the base retriever (a frozen Sentence-BERT encoder). Utilizing fine-tuned retrievers, such as DPR, could partially alleviate this issue.

The second failure scenario occurs when the retrieval verification (RV) module underperforms. Ideally, the RV module should prioritize the LLM’s internal knowledge over retrieved knowledge when the RD module underperforms. Hence, in this scenario, we identify cases where the RV incorrectly chooses retrieved knowledge instead of the LLM’s internal knowledge under RD underperformance, resulting in errors. This accounts for 8 out of 22 samples, highlighting that our proposed RV module is not yet robust enough and leaves room for future research on developing a more accurate RV module for ambiguous questions. The remaining 14 out of 22 samples are particularly challenging and difficult for both RAG and closed-book LLMs.

The third failure scenario occurs when the RD module performs well, but the LLM fails to generate sufficiently accurate responses based on the retrieved passages. This accounts for 16 out of 50 samples and highlights an inherent limitation of the LLM rather than our framework, DIVA. Notably, there are no cases where the RV module underperforms when the RD module performs well.

The final failure scenario arises from other factors, such as the limitations of the evaluation metric D-F1, accounting for 12 out of 50 samples.

### C.2 Experiments on Unambiguous Questions

To assess the performance of DIVA on unambiguous questions, we randomly selected 100 unambiguous questions from the NQ dataset. Specifically, we used the AmbigNQ dataset (Min et al., 2020) to identify whether each question was ambiguous or unambiguous. Utilizing the identifier provided in the AmbigNQ dataset, we first isolated

all unambiguous questions and then randomly sampled 100 questions from this subset for our evaluation. In Table 5, our results indicate that the closed-book LLM achieves an EM score of 75, while incorporating the Vanilla RAG framework boosts the EM score to 80. Significantly, DIVA outperforms both the closed-book LLM and Vanilla RAG, achieving QA performance on par with Iterative RAG. As highlighted throughout the paper, DIVA is also twice as efficient as Iterative RAG while delivering comparable performance. These results confirm that DIVA, while tailored for ambiguous queries, also demonstrates strong performance on unambiguous ones, showcasing its broad applicability.

Method	EM
Closed-book LLM	75.0
Vanilla RAG	80.0
Iterative RAG	<b>83.0</b>
DIVA	<b>83.0</b>

Table 5: QA performance on unambiguous questions.

Additionally, we assessed the effectiveness of DIVA’s Retrieval Diversification (RD) module on unambiguous queries using Recall@5. In Table 6, Vanilla RAG (row 1) refers to the baseline approach, where passages are retrieved based on a given question. + RD (Ours) applies our RD method to the baseline, incorporating pseudo-interpretations generated by DIVA to diversify retrieval. The results indicate that the RD module does not hinder retrieval performance. In fact, as shown in the table, it significantly improves Recall@5 when compared to Vanilla RAG. These findings demonstrate that RD effectively addresses issues of low-quality retrieval, enhancing performance for both ambiguous and unambiguous queries.

Row	Method	Recall@5
1	Vanilla RAG	84.0
2	+ RD (Ours)	<b>87.0</b>

Table 6: Retrieval accuracy on unambiguous questions.

### C.3 Statistical Significance Test

To verify that DIVA consistently outperforms Vanilla RAG, we conduct a statistical significance test on the D-F1 metric. Given the cost of GPT API calls, we use GPT-3.5-turbo as the backbone model. A t-test is performed based on five experimental



runs. The average D-F1 scores for Vanilla RAG and DIVA are 36.8 and 38.1, respectively. The resulting p-value of the t-statistic is 0.0242, which is significantly below the 0.05 threshold, confirming that DIVA achieves statistically significant improvements over Vanilla RAG.

Due to cost constraints, we were limited to five runs. However, we observed that each additional run led to a gradual decrease in the p-value. This suggests that with more runs, the p-value would likely decrease further, providing even stronger statistical evidence for DIVA’s superiority.

## D Prompts

Table 7 and Table 8 show an example of text prompt for inferring *pseudo-interpretations* (i.e.,  $I_a$  and  $I_p$  in Eqn 3). Table 9 shows an example of text prompt for verifying the retrieved passages (i.e.,  $I_v$  in Eqn 7). Table 10 and 11 show an example of text prompt for response generation in vanilla RAG framework (i.e.,  $I_e$  in Eqn 1 and  $I_g$  in Eqn 2)

Table 7: Example of Prompt  $I_a$ .

Instruction
<p>Your task is to determine which types of ambiguity are related to a given question. Types of ambiguity in a question can be defined as follows:</p> <ol style="list-style-type: none"> <li>1. [AmbSub]: This type of ambiguity arises when the subject of the question is not clear. The subject is the person, place, thing, or idea that is doing or being something. It's the entity about which information is being sought.</li> <li>2. [AmbObj]: This type of ambiguity arises when the object of the question is unclear. The object refers to the entity that the action or state expressed by the verb is directed towards.</li> <li>3. [AmbPred]: This type of ambiguity arises when the predicate of the question is unclear. The predicate is the part of a sentence that tells us what the subject does or is. It includes the verb and everything else that comes after the subject.</li> <li>4. [AmbTime]: This type of ambiguity arises when the time frame of the question is unclear. This can lead to confusion because many actions or states can change over time.</li> <li>5. [AmbLoc]: This type of ambiguity arises when the location referred to in the question is unclear. Many events or entities can exist in different locations, leading to confusion.</li> <li>6. [N/A]: This type of ambiguity arises when there is no ambiguous point in the given question.</li> </ol> <p>Below are some examples that map the question to the types.</p> <p>Question: Who has scored the most goals in international soccer Types: [AmbSub]. The subject "Who" may refer to either men or women.</p> <p>Question: What is the date of the queen's birthday? Types: [AmbObj]. The object "the date of the queen's birthday" may refer to the date of Queen Elizabeth II's birthday or Queen Victoria's birthday.</p> <p>Question: Who appeared in the Wimbledon finals 2017? Types: [AmbPred]. The predicate "appeared" could refer to the tennis players or celebrities in the audience.</p> <p>Question: Where is the u21 euro championships being held? Types: [AmbTime]. You may need to clarify whether it refers to the championships being held in 2015, 2017, or 2019.</p> <p>Question: When is the new iPhone being released? Types: [AmbLoc]. This may need clarification on whether it refers to the release date in the United States, Europe, Asia, or another region.</p>
Few-shot Demos
<p>Given the ambiguous question that can be interpreted in multiple ways, which types of ambiguity are related to the question? Suggest the types and provide reasons for your suggestions. Please use the format of: ##Reason: {reason} ##Answer: {answer}.</p> <p>question: Who is top goalscorer in the world cup?</p> <p>##Reason: The subject "Who" in the question may refer to either men or women, as both men's and women's FIFA World Cups are held. ##Answer: [AmbSub]</p>
Actual Question
<p>Given the ambiguous question that can be interpreted in multiple ways, which types of ambiguity are related to the question? Suggest the types and provide reasons for your suggestions. Please use the format of: ##Reason: {reason} ##Answer: {answer}.</p> <p>question: Who has the highest goals in world football? ##Reason: The subject Who in the question is ambiguous as it may refer to either men or women. The disambiguation clarifies this by specifying the gender. ##Answer: [AmbSub]</p>

Table 8: Example of Prompt  $I_p$ .

Instruction
I will provide an ambiguous question that can have multiple answers based on different possible interpretations. Additionally, I will provide corresponding reasons why the question is ambiguous. Clarify the given question into several disambiguated questions based on the reasons for its ambiguity. Please use the format of: ##Disambiguations: {disambiguations}:
Few-shot Demos
##Question: Who is top goalscorer in the world cup?
##Reason: The subject "Who" in the question may refer to either men or women, as both men's and women's FIFA World Cups are held.
##Disambiguations:
1: Who is the top goalscorer in the men's FIFA world cup?
2: Who is the top goalscorer in the women's FIFA world cup?
Actual Question
##Question: Who has the highest goals in world football?
##Reason: The subject "Who" in the question is ambiguous as it may refer to either men or women. The disambiguation clarifies this by specifying the gender.
##Disambiguations:
1: Which male player has the highest goals in world football?
2: Which female player has the highest goals in world football?

Table 9: Example of Prompt  $I_v$ .

Instruction
Given the question and its relevant passages, determine whether the passage contains the answer to the question. Please answer with Yes or No.
Actual Question
Question: Which male player has the highest goals in world football?
Passage:
[1] List of footballers with the most goals in a single game   This is a list of players with the most goals in a football game...
...
[5] List of men's footballers with 50 or more international goals   In total, 79 male footballers to date have scored at least 50 goals with their national team at senior level ...
Response:
Yes.

Table 10: Example of Prompt  $I_e$ .

Instruction
<p>I will provide ambiguous questions that can have multiple answers based on their different possible interpretations. Clarify the given question into disambiguated questions as many as possible and provide short factoid answers to each question. Subsequently, summarize them into a detailed long-form answer of at least three sentences. Here are some examples.</p>
Few shot demos
<p>Question: Who is top goalscorer in the world cup?</p> <p>Disambiguations:</p> <p>DQ 1: Who is the top goalscorer in the men's FIFA world cup?</p> <p>DA 1: Miroslav Klose</p> <p>DQ 2: Who is the top goalscorer in the women's FIFA world cup?</p> <p>DA 2: Marta</p> <p>Answer: There is a men's and women's FIFA world cup tournament. The the top goalscorer in the men's FIFA world cup is German player, Miroslav Klose. And in the women's division, the top goalscorer is a Brazilian player, most commonly known as Marta.</p>
Actual Question
<p>Context:</p> <p>[1] List of footballers with 500 or more goals   Bican (Slavia Prague), Jimmy Jones (Glenavon), ...</p> <p>...</p> <p>[5] List of men's footballers with 50 or more international goals   In total, 79 male footballers ...</p> <p>Question: Who has the highest goals in world football?</p> <p>Disambiguations:</p> <p>DQ 1: Who has scored the highest number of goals in their football career in history?</p> <p>DA 1: Josef Bican</p> <p>DQ 2: Who has scored the highest number of goals in a single football game?</p> <p>DA 2: Archie Thompson; Shokhan Nooraldin Salihi</p> <p>DQ 3: Who has scored the highest number of international goals in men's football?</p> <p>DA 3: Cristiano Ronaldo</p> <p>DQ 4: Who has scored the highest number of goals in a calendar year in international matches?</p> <p>DA 4: Cristiano Ronaldo</p>



Table 11: Example of Prompt  $I_g$

Instruction
I will provide ambiguous questions that can have multiple answers based on their different possible interpretations. Clarify the given question into disambiguated questions as many as possible and provide short factoid answers to each question. Subsequently, summarize them into a detailed long-form answer of at least three sentences. Here are some examples.
Few shot demos
<p>Question: Who is top goalscorer in the world cup?</p> <p>Disambiguations:</p> <p>DQ 1: Who is the top goalscorer in the men's FIFA world cup?</p> <p>DA 1: Miroslav Klose</p> <p>DQ 2: Who is the top goalscorer in the women's FIFA world cup?</p> <p>DA 2: Marta</p> <p>Answer: There is a men's and women's FIFA world cup tournament. The the top goalscorer in the men's FIFA world cup is German player, Miroslav Klose. And in the women's division, the top goalscorer is a Brazilian player, most commonly known as Marta.</p>
Actual Question
<p>Context:</p> <p>[1] List of footballers with 500 or more goals   Bican (Slavia Prague), Jimmy Jones (Glenavon), ...</p> <p>...</p> <p>[5] List of men's footballers with 50 or more international goals   In total, 79 male footballers ...</p> <p>Question: Who has the highest goals in world football?</p> <p>Disambiguations:</p> <p>DQ 1: Who has scored the highest number of goals in their football career in history?</p> <p>DA 1: Josef Bican</p> <p>DQ 2: Who has scored the highest number of goals in a single football game?</p> <p>DA 2: Archie Thompson; Shokhan Nooraldin Salihi</p> <p>DQ 3: Who has scored the highest number of international goals in men's football?</p> <p>DA 3: Cristiano Ronaldo</p> <p>DQ 4: Who has scored the highest number of goals in a calendar year in international matches?</p> <p>DA 4: Cristiano Ronaldo</p> <p>Answer:</p> <p>The question "Who has the highest goals in world football?" can be interpreted in several ways. If we consider the highest number of goals scored in ...</p>

Table 12: Example of Prompt  $I_1$

Instruction
I will provide ambiguous questions that have multiple answers regarding different aspects of the question. Your task is to generate an answer that includes as many aspects as possible from the ambiguous questions.
Few shot demos
<p>Question: Who is top goalscorer in the world cup?</p> <p>Given the question, generate a comprehensive long-form answer.</p> <p>Final Answer: There is a men's and women's FIFA world cup tournament. The the top goalscorer in the men's FIFA world cup is German player, Miroslav Klose. And in the women's division, the top goalscorer is a Brazilian player, most commonly known as Marta.</p>
Actual Question
<p>Question: Who has the highest goals in world football?</p> <p>Given the question, generate a comprehensive long-form answer.</p> <p>Final Answer: The highest goals in world football can be interpreted in different ways. If we are talking about the highest number of goals scored in a professional football career, the record belongs to Josef Bican, who scored an estimated 805 goals in competitive ...</p>