Real-time Verification and Refinement of Language Model Text Generation

Anonymous ACL submission

Abstract

Large Language Models (LLMs) have shown remarkable performance across a wide range of natural language processing tasks. However, a critical challenge remains in that they sometimes generate the factually incorrect answers, known as hallucination. To address this, while many previous work has focused on identifying the errors in their generation and further refining them, they are slow in deployment since they are designed to verify the response from LLMs only after their entire generation (from the first to last tokens) is done. Further, we observe that once LLMs generate incorrect tokens early on, there is a higher likelihood that subsequent tokens will also be factually incorrect. To this end, in this work, we propose **Streaming-VR** (**Streaming V**erification and **R**efinement), a novel approach designed to enhance the efficiency of verification and refinement of LLM outputs. Specifically, the proposed Streaming-VR enables on-the-fly verification and correction of tokens as they are being generated, similar to a streaming process, ensuring that each subset of tokens is checked and refined in real-time by another LLM as the LLM constructs its response. Through comprehensive evaluations on multiple datasets, we demonstrate that our approach not only enhances the factual accuracy of LLMs, but also offers a more efficient solution compared to prior refinement methods.

1 Introduction

Large Language Models (LLMs) (Achiam et al., 2023; Jiang et al., 2023a; Dubey et al., 2024) have shown significant advancements due to their capabilities in various tasks like question answering (QA) (Yang et al., 2018; Kwiatkowski et al., 2019; Fan et al., 2019; Min et al., 2020) and more complicated QA tasks with the help of information retrieval (IR) (Xiong et al., 2020; Karpukhin et al., 2020; Ni et al., 2021) for real-world applications. However, LLMs still face notable limitations like hallucinations, mainly due to the inherent complexity and difficulties of natural languages (Rawte et al., 2023) and the wrong application and generalization of memorized or retrieved knowledge (Jiang et al., 2024; Xu et al., 2024). To elaborate, models tend to produce factually incorrect or misleading content, which is exacerbated by their lack of access to real-world knowledge sources.

Previous approaches have sought to mitigate these inaccuracies by augmenting LLMs with external knowledge sources (Guu et al., 2020; Lewis et al., 2020; Luo et al., 2023). However, these methods often face challenges in maintaining faithfulness, as they may retrieve information that is either ungrounded or irrelevant to the context. Additionally, LLMs remain constrained by their black-box nature, which lacks explainability and transparency, making it difficult to interpret the reasoning or trust their outputs, particularly in critical applications. The resource-intensive nature of LLMs poses accessibility challenges, limiting their deployment in resource-constrained environments. These limitations highlight the need for more efficient and transparent methods to enhance the factual accuracy and reliability of LLM outputs in complex QA tasks. To this end, in the realm of error identification and verification, recent research has highlighted the challenges LLMs face in accurately detecting and correcting mistakes (Peng et al., 2023).

However, traditional methods primarily focus on identifying inaccuracies and correcting misinformation only after the entire answer has been generated, leading to inefficiencies and increased time consumption. Specifically, LLMs generate text sequentially, predicting one token at a time based on the preceding context. An early error in this sequence can trigger a cascade of inaccuracies in subsequent tokens, as each token is influenced by its predecessors. This error propagation underscores the necessity for intermediate corrections to prevent further inaccuracies throughout the response.

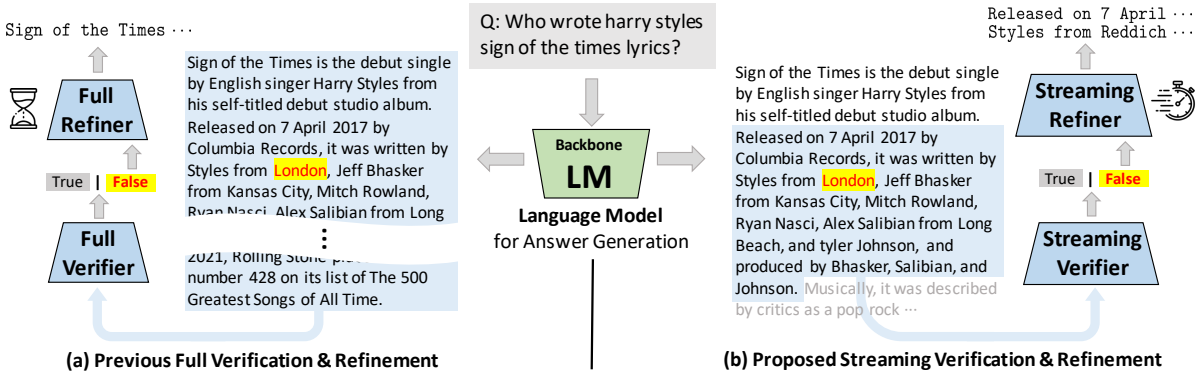


Figure 1: **(a)**: Previous verify-and-refine framework after the entire answer generation. **(b)**: Our proposed method, **Streaming-VR**, that verifies intermediate answers in sentence-level and refine them if identified as error, with remarkable efficiency.

After generation, the model must re-evaluate the entire text to identify and rectify misinformation (see Figure 1 (a)). This process is computationally expensive and time-consuming, especially when it involves examining the entire sequence rather than just the sequence containing incorrect tokens, particularly when errors are complex or numerous due to earlier propagation in the responses..

In this work, we propose Streaming-VR (**Streaming** **V**erification and **R**efinement), a method designed to address the issue of error propagation in LLM-generated text. As visually depicted in Figure 1 (b), Streaming-VR evaluates model-generated answers in real-time, identifying and correcting only the erroneous sentences as they are being produced. By employing an external verification model, Streaming-VR verifies errors during the generation process, detects inaccuracies in newly generated sequence of tokens, and promptly corrects them. Because rectification occurs immediately after verification and runs concurrently with text generation, Streaming-VR significantly enhances efficiency and improves the factual accuracy of model outputs. Our findings also indicate that when LLMs produce incorrect tokens early in the sequence, it substantially increases the likelihood of subsequent tokens being factually inaccurate, underscoring the critical importance of employing Streaming-VR to mitigate error propagation. Also, we validate the effectiveness and efficiency of Streaming-VR experimentally for two benchmark datasets, ASQA (Stelmakh et al., 2022) and QuoteSum (Schuster et al., 2024), using Mistral 7B (Jiang et al., 2023a), LLaMA-3.1 8B (Dubey et al., 2024), GPT-4o (Achiam et al., 2023), the language models with various sizes.

2 Related Work

Large Language Models Recent advancements in Language Models (LMs) (Radford, 2018; Devlin et al., 2019; Liu et al., 2019; Raffel et al., 2023) and LLMs with billions of parameters, have led to significant improvements in performance across various natural language tasks. Since LMs cannot memorize or learn every real-world knowledge, several studies have explored methods to enhance their capabilities by leveraging external knowledge sources like retrieval-augmented generation (Lewis et al., 2020), for knowledge-intensive tasks. Despite the assistance of external knowledge, models often generate incorrect answers due to the failure of factual recall (Jiang et al., 2024) as they may not succeed in retrieving or applying the relevant knowledge appropriately, and generalizing memorized knowledge accurately.

Language Model Verification and Refinement To tackle this issue, recent research has focused on verifying the relevance and accuracy of retrieved knowledge using a separate verifier (Baek et al., 2023). Additionally, generating answers through retrieval on-demand retrieval of external information using special retrieval token, followed by critiquing the output to improve it has been explored (Asai et al., 2024). Also, Jeong et al. (2024) have developed more efficient adaptive retrieval method using an external query complexity classifier. A dynamic retrieval process that determines when and what to retrieve during answer generation (Jiang et al., 2023b) has shown improvements in knowledge-intensive tasks, as the retrieve-and-generate approach is inadequate for tasks requiring the generation of lengthy texts, where continuous integration of information throughout the process is essential.

Other than the misinformation induced by wrong knowledge, LLM itself often generates plausible but incorrect texts (Zhang et al., 2024) (*i.e.*, hallucination). Thus, evaluating the factuality (Thorne et al., 2018; Min et al., 2023) of LLM outputs correcting inaccuracies has emerged as an important topic. Various approaches explore methods to enhance the factual accuracy of model responses and

develop robust fact-checking or answer-verifying models. For instance, Dhuliawala et al. (2024) generates a series of independent questions to check the factual claims made in the model response, followed by synthesizing the answers from the verification step. Beyond evaluating or verifying the faithfulness of LLM answers, answer-correction has also become a prominent area of focus in various fields. Iterative refinement is well known to be helpful for improving generative contents of natural language (Madaan et al., 2024) and code (Faltings et al., 2021; Yasunaga et al., 2021) autonomously, but limited to the final outcome.

On the other hand, Lightman et al. (2023) demonstrates the effectiveness of process supervision by focusing on each step of the reasoning process, and allowing the model to identify and correct errors in the middle. It emphasizes the importance of intermediate verification in complex multi-step reasoning tasks like mathematical problem solving where a single error can derail the entire answer. Also, Welleck et al. (2023) employs an online training procedure for a separate corrector to learn from feedback on intermediate outputs. Nevertheless, LMs are capable of correcting errors when their exact locations are identified (Tyen et al., 2024), which poses a bottleneck in improving self-correction capabilities. Furthermore, Huang et al. (2024) have demonstrated through experimental analyses that current LLMs struggle to self-correct their reasoning without external feedback, often resulting in degraded performance after attempting self-correction. On the other hand, Cobbe et al. (2021); Wang et al. (2023) utilize a trained critique model or verifier to correct errors on responses through their feedback. In addition, Gou et al. (2024) show that verification and correction can be done effectively by interacting with diverse external tools. By integrating all the inspirations from these works, we propose a novel method that refines the specific intermediate sentence of an answer identified as incorrect, with higher efficiency.

3 Method

3.1 Preliminaries

We begin with preliminaries, formally explaining Large Language Models and traditional verify-and-refine approach, Full Verification and Refinement.

Large Language Models Let us define the process of generating an answer $\hat{\boldsymbol{a}}$ to a given question $\boldsymbol{q}$ as a function: $\hat{\boldsymbol{a}} = \texttt{LLM}(\boldsymbol{q})$. The quality or accuracy of this generated answer is assessed using a scoring function, denoted as $\texttt{score}(\hat{\boldsymbol{a}}, \boldsymbol{a})$, where $\boldsymbol{a}$ represents the correct reference answer.

In QA systems that incorporate external knowledge, such as in retrieval-augmented generation (RAG), or examples, as in in-context learning (ICL), the answering process differs slightly. Formally, let $\boldsymbol{d}$ denote the external knowledge or example retrieved from the source $\mathcal{D}$. To this end, the retrieval is performed using a dedicated retrieval model $\texttt{Retriever}$, for a given query $\boldsymbol{q}$, defined as: $\boldsymbol{d} = \texttt{Retriever}(\boldsymbol{q}; \mathcal{D})$. This process involves ranking the retrieved data based on its relevance or similarity to the given query. After the related documents are retrieved for RAG or ICL, we now incorporate them as input to the LLMs as: $\hat{\boldsymbol{a}} = \texttt{LLM}(\boldsymbol{q}, \boldsymbol{d})$. For more detailed explanations, please refer to the section of implementation details in Appendix A.1. For the real-time sentence-level verification and refinement, we also analyze the individual sentences in answer. To elaborate, an answer $\hat{\boldsymbol{a}}$ is structured as a sequence of n sentences, expressed as $\hat{\boldsymbol{a}} = [s_1, s_2, \cdots, s_n]$, where the notation $[\cdot]$ signifies concatenation in the specified order. To facilitate real-time correction of incorrect sentences within intermediate answers, we define the intermediate answer at a certain step t ($t \geq 1$) as $\hat{\boldsymbol{a}}_{\leq t} = [s_1, \cdots, s_t]$ containing t sentences in total. Note that this can also be expressed as $\hat{\boldsymbol{a}}_{\leq t} = [\hat{\boldsymbol{a}}_{\leq t-1}, s_t]$, where s_t is the most recently generated sentence in streaming setup. We initialize $\hat{\boldsymbol{a}}_{\leq 0}$ as an empty string for coherence.

Full-VR The traditional approach for verifying and refining LLM answers, namely Full-VR (Full Verification and Refinement), is the most common strategy for improving them just by re-generating the entire responses if identified to be incorrect. While many previous works achieve significant improvements through supplementary techniques, we focus solely on the vanilla Full-VR setting, to rectify the answer containing at least one erroneous token in that context, for a direct comparison with our method. Using the notations defined above, the overall Full-VR pipeline is expressed as follow for a given query $\boldsymbol{q}$ and its answer $\hat{\boldsymbol{a}} = \texttt{LLM}(\boldsymbol{q})$:

$$\tilde{\boldsymbol{a}} = \begin{cases} \hat{\boldsymbol{a}} & \textit{if} \quad o = \texttt{True} \\ \texttt{Refiner}(\hat{\boldsymbol{a}}) & \textit{if} \quad o = \texttt{False} \end{cases}$$

where $o = \texttt{Verifier}(\hat{\boldsymbol{a}})$ is the verification output, and $\tilde{\boldsymbol{a}}$ is the final output of Full-VR, respectively.

And for a fair comparison by matching the target answers to refine, we utilize the sentence-level verification result $o_t = \texttt{Verifier}(\hat{\boldsymbol{a}}_{\leq t})$ for the t-th sentence, from Streaming-VR, as an answer is considered incorrect if at least one of its constituent sentences is wrong. Formally, an answer composed of n sentences is verified as follows, where $\wedge$ denotes the logical AND operator.

$$o = \texttt{Verifier}(\hat{\boldsymbol{a}}) = \bigwedge_{i=1}^{n} o_i = \bigwedge_{i=1}^{n} \texttt{Verifier}([\hat{\boldsymbol{a}}_{\leq i-1}, s_i])$$

3.2 Streaming-VR: Streaming Verification and Refinement

Our approach is structured in the following steps during the generation of answers: 1) Generation of Response 2) Streaming-Verfication and 3) Streaming-Refinement of intermediate answers. We formulate the overall framework of Streaming-VR as follow for a given query $\boldsymbol{q}$ and the t-th sentence s_t in its answer, such that $s_t \in \texttt{LLM}(\boldsymbol{q})$:

$$\tilde{s}_t = \begin{cases} s_t & \textit{if} \quad o_t = \texttt{True} \\ \texttt{Refiner}(s_t) & \textit{if} \quad o_t = \texttt{False} \end{cases}$$

where $o_t = \texttt{Verifier}([\hat{\boldsymbol{a}}_{\leq t-1}, s_t])$ is the verification output, and $\tilde{s}_t$ is the new sentence output of Streaming-VR at a certain step t. Note that the refinement model, Refiner takes into account the whole context of previously verified and (may have been) refined sentences, $\tilde{\boldsymbol{a}}_{\leq t-1} = [\tilde{s}_1, \cdots, \tilde{s}_{t-1}]$. After processing all the sentences by Streaming-VR, the final refined answer output should be in the form as follow: $\tilde{\boldsymbol{a}} = [\tilde{s}_1, \cdots, \tilde{s}_n]$.

The verification of answer is mainly based on the verifier output, $o_t = \texttt{Verifier}(\hat{\boldsymbol{a}}_{\leq t})$ such that $o_t \in \{\texttt{True}, \texttt{False}\}$. We employ a finetuned LLM to return the output identifying whether the input is True or False by checking the factuality of the generated answers in sentence-level. To this end, we augment true- and false-labeled sentences for verifier training, as there is no proper sentence-level question answering dataset. The augmented sentences are made from the provided reference answer data by rephrasing it for True and adding wrong information for False by GPT-4o (Achiam et al., 2023). To suit real-time verification scenarios, we split the answer data into individual sentences using NLTK (Bird and Loper, 2004) and concatenate them in a specific order to form intermediate answers as $\{\hat{\boldsymbol{a}}_{\leq 1}, \cdots, \hat{\boldsymbol{a}}_{\leq t}\}$, ensuring that False labeled sentences can only appear at the end and never in the middle. Note that a special token for sentence separation, [SEP] is inserted right before the last sentence in each intermediate answer as $[s_1, s_2, \cdots, \texttt{[SEP]}, s_t]$ for a certain stage t. This setup allows a model to be trained to verify the last sentence along with the context from the preceding True-labeled paragraph. The descriptions regarding the implementation of the streaming verifiers are provided with more detail in Appendix A.1.

The refinement of each answer is contingent upon the verification output, with careful consideration given to the paragraph preceding the target sentence that requires correction. In this process, we adopt LLMs akin to backbone models refraining from utilizing any additional techniques. To facilitate a real-time scenario with conventional language models, we provide the entire prompt given for answering the test query to the refinement model. Additionally, we only include the retrieved passages or few-shot examples given to the generation prompt, without incorporating any extra information from external knowledge sources for refinement. This strategy ensures that the contextual information relevant to the intermediate generation processes is fully incorporated. Furthermore, as the intermediate answers are refined, they must be updated to reflect the newly refined preceding sentences, thereby enabling a continuous and coherent streaming refinement process.

4 Experiments

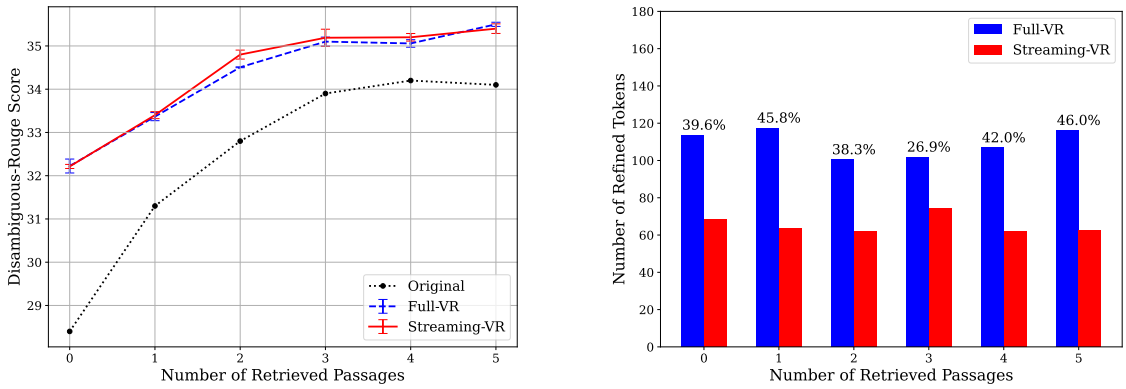
4.1 Datasets and Evaluation Metrics

We leverage two different datasets to evaluate the effect of Streaming-VR especially for multi-answer questions which require well-grounded responses to assess the trustworthiness of QA systems.

ASQA (Stelmakh et al., 2022) is a challenging dataset that bridges the gap between factoid and long-form QA tasks as it requires multiple answers for ambiguous questions. It is composed of 4,353 and 948 questions in the train and dev sets, respectively, while the test set is not publicly available. Thus, we use the dev set as our test set here. ASQA provides the reference long-form answers for every questions which are originated from AmbigQA (Min et al., 2020), the ambiguous questions subset of questions from NQ (Kwiatkowski et al., 2019). In this paper, to evaluate the quantitative performance of methods on ASQA, we follow the official metrics and report: Disambiguous-Rouge

Table 1: Results of Streaming-Verification by **Mistral 7B** and Refinement by **GPT-4o** on **ASQA** for three different backbone models. $\mathcal{T}$ indicates the number of tokens generated for refinement. We report the average values over three-runs.

ASQA										
	Closed-Book					Open-Book with 5 Passages				
Method	ROUGE-L	Disambig-F1	DR	$\mathcal{T}$	Efficiency ↑	ROUGE-L	Disambig-F1	DR	$\mathcal{T}$	Efficiency ↑
Mistral 7B	33.6	20.7	26.4	–	–	36.4	31.2	33.7	–	–
+ Full-VR	35.3	29.6	32.3	113.6	39.6%	36.6	33.9	35.2	101.8	26.9%
+ **Streaming-VR**	35.2	29.6	32.3	68.6		36.9	33.7	35.3	74.4	
LLaMA-3.1 8B	34.0	23.7	28.4	–	–	36.6	31.7	34.1	–	–
+ Full-VR	35.2	29.4	32.2	117.4	45.8%	37.0	34.2	35.6	106.8	42.1%
+ **Streaming-VR**	35.3	29.4	32.2	63.6		36.8	34.0	35.4	61.9	
GPT-4o	36.6	34.8	35.7	–	–	37.1	35.0	36.0	–	–
+ Full-VR	35.2	29.6	32.3	100.4	38.3%	37.0	33.9	35.4	116.1	46.0%
+ **Streaming-VR**	35.3	29.4	32.2	61.9		36.9	33.9	35.4	62.7	



(a) Performance according to the Number of Retrieved Passages **(b) Number of Refined Tokens according to the Number of Retrieved Passages**

Figure 2: Result of Streaming-Verification by **Mistral 7B** and Refinement by **GPT-4o** for the answers generated by **LLaMA-3.1 8B** on **ASQA** dataset according to the number of retrieved passages. **(a)**: As more passages are given, the DR score increases as Streaming-VR shows competitive performance to Full-VR . **(b)**: Streaming-VR is more efficient compared to Full-VR while maintaining its competitive answering performance. The numbers on top of the bars are their efficiency values.

(DR) as the overall score which combines ROUGE-L (Lin, 2004) for text quality and Disambig-F1 (QA score based on RoBERTa large (Liu et al., 2019)) for factual correctness.

QuoteSum (Schuster et al., 2024) is also a difficult question answering dataset for Semi-Extractive Multi-source Question Answering (SEMQA), a task designed to assess the comprehensive answering ability by summarizing information from multiple sources. To be specific, SEMQA requires models to generate a comprehensive response that integrate verbatim factual spans extracted from input sources along with supplementary non-factual text connecting them, thereby ensuring a cohesive answer. QuoteSum is made up of 4,009 semi-extractive answers to 1,376 unique questions from PAQ (Lewis et al., 2021) and NQ. For the quantitative evaluation on QuoteSum, we follow the official metrics and report: ROUGE-L, Sem-F1 for answer extraction quality, and overall SEMQA score where they do not require any model-based evaluations.

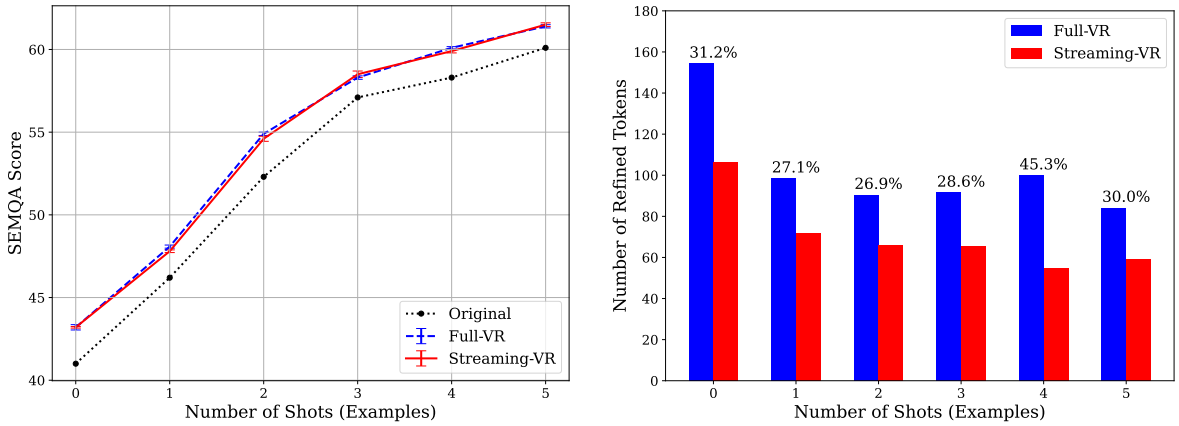
In addition to evaluating answer quality and factual accuracy, we also measure token count to assess the efficiency of each method. This metric is crucial as the number of refined tokens directly affects to the LLM user's waiting time for response corrections. To quantify this, we define the efficiency of Streaming-VR relative to Full-VR, taking a cue from the thermal efficiency in thermodynamics, which is formulated as: (Efficiency) $:= \frac{\text{benefit}}{\text{cost}} = 1 - \frac{\mathcal{T}_S}{\mathcal{T}_F}$, where $\mathcal{T}_S$ and $\mathcal{T}_F$ represent the average number of refined tokens per answer for Streaming-VR and Full-VR, respectively.

4.2 Experimental Results and Analyses

Streaming-VR delivers higher efficiency while maintaining its performance. We perform a series of experiments on the ASQA and QuoteSum datasets to quantitatively evaluate the efficiency and effectiveness of two verification and refinement approaches: Streaming-VR and Full-VR. To facilitate this comparison, we first segment the answers generated by the model for each test query into individual sentences, thereby treating these sentences arranged in a specific sequential order, as distinct intermediate answers. Following this segmentation, we employ Streaming-VR to each of intermediate answers. It allows us to verify and refine the answers in real-time, enabling a dynamic adjustment process as the responses are generated. In contrast, we utilize Full-VR as a baseline method, which verifies and refines the entire answer only

Table 2: Results of Streaming-Verification by **Mistral 7B** and Refinement by **GPT-4o** on **QuoteSum** for three different backbone models. $\mathcal{T}$ indicates the number of tokens generated for refinement. We report the average values over three-runs.

QuoteSum										
	Zero-Shot					Five-Shots				
Method	ROUGE-L	Sem-F1	SEMQA	$\mathcal{T}$	Efficiency ↑	ROUGE-L	Sem-F1	SEMQA	$\mathcal{T}$	Efficiency ↑
Mistral 7B	37.5	39.0	38.2	–	–	46.8	51.8	50.1	–	–
+ Full-VR	38.1	39.0	38.5	101.3	25.8%	57.6	49.0	53.1	72.5	24.3%
+ **Streaming-VR**	37.9	39.0	38.4	75.2		57.5	48.9	52.9	54.9	
LLaMA-3.1 8B	43.3	38.9	41.0	–	–	59.1	61.2	60.1	–	–
+ Full-VR	47.6	39.0	43.1	154.3	31.2%	60.7	62.1	61.4	84.1	30.0%
+ **Streaming-VR**	47.5	39.0	43.0	106.1		60.7	62.3	61.5	58.9	
GPT-4o	60.3	39.0	48.5	–	–	65.8	54.7	60.0	–	–
+ Full-VR	60.2	39.0	48.5	60.7	26.7%	65.8	54.7	60.0	78.9	42.2%
+ **Streaming-VR**	60.0	39.0	48.4	44.5		65.3	54.7	59.8	45.6	



(a) Performance according to the Number of Shots

(b) Number of Refined Tokens according to the Number of Shots

Figure 3: Result of Streaming-Verification by **Mistral 7B** and Refinement by **GPT-4o** for the answers generated by **LLaMA-3.1 8B** on **QuoteSum** dataset according to the number of given examples (shots). **(a)**: As more examples are given, the SEMQA score increases as Streaming-VR shows competitive performance to Full-VR . **(b)**: Streaming-VR is more efficient compared to Full-VR while maintaining its competitive answering performance. The numbers on top of the bars are their efficiency values.

after the complete sequence has been generated, processing the output from the first to the last token in a single pass. By contrasting the performance of Streaming-VR against Full-VR, we aim to elucidate the advantages of real-time refinement in enhancing answer quality and efficiency.

The main results on the ASQA are summarized in Table 1. Both Full-VR and Streaming-VR employ Mistral 7B as the verifier and GPT-4o as the refiner across three different backbone models (Mistral 7B, LLaMA-3.1 8B, and GPT-4o) for answer generation, as indicated in the method column. Across all backbones, the final outcomes after verification converge to similar scores, indicating that the overall quality and faithfulness of the answers are largely determined by the refinement model. However, we observe a notable performance decline when GPT-4o is used as the backbone for answer generation. Both Full-VR and Streaming-VR with GPT-4o lead to significant drops in Disambig-F1, a key metric for assessing the informativeness of long-form answers. This suggests that GPT-4o, which already generates high-quality answers, may be susceptible to overcorrection during the refinement process, reducing the overall effectiveness of the responses. This highlights a broader trend: refining answers with the same model used for generation, even if it is a large model like GPT-4o, may not enhance performance and can even degrade it. In terms of efficiency, Streaming-VR offers substantial advantages over Full-VR across all models and both closed-book and open-book settings. While Full-VR refines the entire response, requiring more tokens, Streaming-VR operates at the sentence level, refining only those sentences identified as inaccurate, resulting in significantly fewer tokens being used. Consequently, Streaming-VR consistently achieves greater token efficiency than Full-VR, with token savings ranging from 26.9% to 46%, depending on the model and setting. These results confirm that our approach ensures higher efficiency by preventing error propagation during answer generation.

Additional experimental results visualized in Figure 2, show Streaming-VR's competitive performance compared to Full-VR across various numbers of retrieved passages for answer generation. Streaming-VR consistently outperforms the original LLM-generated answers without refinement and achieves comparable results to Full-VR.

Table 3: Result of Streaming-VR with different verifiers, **LLaMA-3.1 8B** as a backbone, and **GPT-4o** as a refiner. Models for `Verifier` are indicated as Streaming-{`Verifier`}G for conciseness, where M, L and G stand for Mistral-7B, LLaMA-3.1 8B and GPT-4o, respectively. Here, R-L, Dis-F1 denote ROUGE-L and Disambig-F1, respectively. We report the average values over three-runs for Streaming-VR.

ASQA						
	Closed-Book			**Open-Book with 5 Passages**		
Method	R-L	Dis-F1	DR	R-L	Dis-F1	DR
LLaMA-3.1 8B	34.0	23.7	28.4	36.6	31.7	34.1
+ Streaming-MG	35.3	29.4	32.2	36.8	34.0	35.4
+ Streaming-LG	35.2	28.3	31.6	36.8	33.8	35.3
QuoteSum						
	Zero-Shot			**Five-Shots**		
Method	R-L	Sem-F1	SEMQA	R-L	Sem-F1	SEMQA
LLaMA-3.1 8B	43.3	38.9	41.0	59.1	61.2	60.1
+ Streaming-MG	47.5	39.0	43.0	60.7	62.3	61.5
+ Streaming-LG	47.7	39.0	43.1	61.0	62.3	61.6

Table 4: Result of Streaming-VR with different refiners, **LLaMA-3.1 8B** as a backbone, and **Mistral 7B** as a verifier. Models for `Refiner` are indicated as Streaming-M{`Refiner`} for conciseness, where M, L and G stand for Mistral-7B, LLaMA-3.1 8B and GPT-4o, respectively. Here, R-L, Dis-F1 denote ROUGE-L and Disambig-F1, respectively. We report the average values over three-runs for Streaming-VR.

ASQA						
	Closed-Book			**Open-Book with 5 Passages**		
Method	R-L	Dis-F1	DR	R-L	Dis-F1	DR
LLaMA-3.1 8B	34.0	23.7	28.4	36.6	31.7	34.1
+ Streaming-MM	34.5	23.7	28.6	36.2	30.5	33.2
+ Streaming-ML	34.2	24.3	28.8	36.8	31.1	33.8
+ Streaming-MG	35.3	29.4	32.2	36.8	34.0	35.4
+ *Self-VR*	34.2	23.3	28.2	36.6	31.1	33.7
QuoteSum						
	Zero-Shot			**Five-Shots**		
Method	R-L	Sem-F1	SEMQA	R-L	Sem-F1	SEMQA
LLaMA-3.1 8B	43.3	38.9	41.0	59.1	61.2	60.1
+ Streaming-MM	39.6	38.9	39.3	58.0	61.2	59.6
+ Streaming-ML	45.2	39.0	41.9	59.9	61.7	60.8
+ Streaming-MG	47.5	39.0	43.0	60.7	62.3	61.5
+ *Self-VR*	42.4	38.9	40.6	57.4	61.2	59.3

The results on QuoteSum are in Table 2, where we conduct the experiments with the models and setting identical to that of ASQA. Notably, the GPT-4o backbone model shows a crucial observation that the refinement process, whether through Full-VR or Streaming-VR, does not yield any significant improvement in performance. This is evident from the fact that the SEMQA scores remain virtually the same even after the refinement. Specifically, in both the zero-shot and five-shots settings, GPT-4o's Full-VR and Streaming-VR results are nearly identical, indicating that refining the answers generated by GPT-4o does not enhance their quality, similar to the results on ASQA. This results again suggests that GPT-4o already produces high-quality answers, and further refinement may be redundant to performance. Further experiment results with various numbers of retrieved example passages for answer generation is visualized in Figure 3. It also illustrates that Streaming-VR delivers results on par with Full-VR across all example shot counts, offering marginal performance improvements over the unrefined original outputs of language model. In summary for both datasets, while refinement methods can enhance answer quality in some cases, they may lead to performance degradation when applied to models like GPT-4o that already generate high-quality outputs. Nonetheless, Streaming-VR consistently outperforms Full-VR in terms of token efficiency ranging from 24.3% to 42.2% while preserving the competitive performance, making it a more practical approach for scenarios where computational resources are a key concern.

You don't need bigger verification models The results in Table 3 clearly demonstrate that a verifier model does not need to be large to be effective. In both ASQA and QuoteSum tasks, Mistral 7B performs comparably to larger verification models, presenting that smaller models can still deliver robust performance improvements. For ASQA, specifically, verification with Mistral 7B (Streaming-MG) achieves slightly better overall performance compared to verification with LLaMA-3.1 8B (Streaming-LG). Thus, small verifiers can improve performance across multiple metrics without incurring computational overhead of larger models. This finding suggests that smaller models can efficiently serve as verifiers, offering substantial improvements in task performance with the same refinement model while maintaining a lower resource footprint. Relying on a smaller verifier allows for a more resource-efficient model pipeline, which is particularly valuable in real-world applications where computational resources are often limited. Since the parameter size difference between the two verifiers is relatively small (approximately 0.73B), we further analyze the impact of self-verification followed by self-refinement in §4.2. The results underscore that an external verifier is still necessary but does not need to be stronger than the backbone model; a similarly sized or smaller verifier suffices. We leave the exploration of a wider range of small verifier sizes, with the goal of enhancing computational efficiency in streaming verification, for future work.

You need bigger refinement models The results in Table 4 strongly emphasize the importance of having a larger, more sophisticated model specifically for the refinement step after verifica-

tion, even when the verifier model is relatively small. For ASQA dataset, introducing Mistral 7B as both the verifier and refiner (Streaming-MM) offers marginal improvements only for textual quality (ROUGE-L) in closed-book setting, and struggles to effectively handle disambiguation in more complex tasks with 5 retrieved passages for open-book setting. However, the utilization of larger models as refiners produces significant improvements. When LLaMA-3.1 8B is employed as the refiner (Streaming-ML), there is a minor increase in Disambig-F1 for closed-book setting, but handling multiple passages is still critical. The most substantial performance boost comes from using GPT-4o as the refiner (Streaming-MG) as you may expected intuitively. The results for both closed- and open-book setting underscore the effectiveness of using GPT-4o's sophisticated reasoning capabilities during the refinement phase. It clearly demonstrates that even though the verifier model is relatively small (Mistral-7B), the refinement model's capacity plays a critical role in improving the system's overall performance. The superior results from GPT-4o confirm that a refinement model larger than the backbone model is essential for handling complex disambiguation and multi-passage reasoning, which smaller models are unable to manage as effectively. This trend continues in the results on QuoteSum dataset as well. Streaming-ML offers improvements for both zero- and five-shot settings, while Streaming-MM degrades the overall answer quality. This illustrates that smaller or same size models struggle for answer correction. Yet again, the strongest performance is observed with GPT-4o as the refiner (Streaming-MG), and it demonstrates that even in the absence of task-specific examples, a larger refiner like GPT-4o is crucial for generating high-quality, coherent answers.

LLMs still struggle with intrinsic self-correction Additionally, we conduct some experiments to evaluate the efficacy of self-verification and self-refinement within the Streaming-VR pipeline, utilizing only LLaMA-3.1 8B for backbone, verifier and refiner models. In Table 4, the rows of *Self-VR* (Self-Verification and Refinement; *i.e.,* Streaming-LL) illustrate that LLMs continue to face challenges with intrinsic self-correction with some performance drops. This result strengthens the conclusions drawn by Huang et al. (2024), which also have demonstrated that intrinsic self-correction, an approach that model attempts to rectify its initial responses using only its inherent capabilities without external feedback, degrades the response quality.

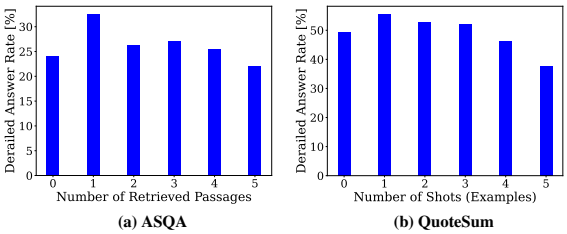


Figure 4: The ratio of derailed answers to incorrect answers. **(a)**: The rate of derailed answers on **ASQA**. **(b)**: The rate of derailed answers on **QuoteSum**.

Any error in the middle derails the entire answer As Zhang et al. (2024) point out that the mistakes or hallucinations in the middle of answer can skew the whole response, we report the statistics of model-generated answers with the rate of derailed answers on each dataset. Specifically, the answers are generated by LLaMA-3.1 8B and verified the finetuned streaming verifier as before. The rate of derailed answers is ratio of the number of 'answers composed of false sentences in sequence to the last sentence' to 'false answers where any answers are identified as false if one of their sentences is false'. Results in Figure 4 for ASQA and QuoteSum are 26.3% and 48.9% in average across different settings for RAG and ICL, respectively. Therefore, they highlight the importance of Streaming-VR to prevent derailed responses.

5 Conclusion

In this paper, we introduce Streaming-VR, a novel approach aimed at improving the efficiency of verification and refinement in language model text generation. Unlike traditional methods that detect and correct errors and misinformation only after the entire response is generated, Streaming-VR performs real-time verification and correction of erroneous token sequences as they are being produced. This prevents error propagation in the early stage and significantly enhances the efficiency of answer refinement. Additionally, unlike previous approaches that either rely on self-verification followed by self-correction or correct only the final responses using external tools, our method verifies and refines hallucinated answers immediately during generation to reduce the likelihood of generating further errors. Extensive experiments for two different QA datasets have clearly demonstrated that Streaming-VR consistently achieves remarkably higher efficiency without compromising response quality.

Limitations

Despite the improvements introduced by our method, Streaming-VR, which enhances both the efficiency and effectiveness of verification and refinement in language model text generation by intervening during intermediate answer generation, there remain promising opportunities for enhancing the answer verifier. Specifically, the primary challenge is the lack of dedicated datasets for answer verification, particularly those suited for real-time scenarios. To address this, we automatically augmented data by paraphrasing sentences or introducing errors by an LLM. However, while effective, this approach carries the risk of mislabeling. Therefore, future work could focus on developing new datasets that are carefully annotated with a diverse range of answers ensuring more accurate verification and reducing the risk of incorrect labeling. Additionally, we can further extend these datasets to include fine-grained labels for multiple classes, rather than just binary ones, to accommodate different types of errors and apply adaptive strategies for subsequent refinement after verification.

Ethics Statement

In our research, we use publicly available question-answering (QA) datasets to evaluate the effectiveness and applicability of Streaming-VR in real-world scenarios. However, these datasets encompass a wide variety of questions, which necessitates careful consideration of potentially offensive or harmful inputs. The language model we employ may inadvertently reflect biases embedded in its training data, resulting in outputs that perpetuate racism, sexism, or other forms of discrimination. Such biases can manifest even in contexts that appear neutral, highlighting the need for proactive bias detection and mitigation strategies. Moreover, harmful inputs might lead to the retrieval of offensive information or the generation of inappropriate responses by the language models. This presents a significant risk that we must recognize and address. To mitigate these issues, it is crucial to develop methods for detecting and managing offensive, inappropriate, or biased content in both user inputs and the documents retrieved within our retrieval-augmented framework. We view this as a critical area for future research because minimizing the risk of biased or harmful outputs is essential for the safe and ethical deployment of QA systems.

References

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

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2024. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations*.

Jinheon Baek, Soyeong Jeong, Minki Kang, Jong Park, and Sung Hwang. 2023. Knowledge-augmented language model verification. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1720–1736, Singapore. Association for Computational Linguistics.

Steven Bird and Edward Loper. 2004. NLTK: The natural language toolkit. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pages 214–217, Barcelona, Spain. Association for Computational Linguistics.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. 2024. Chain-of-verification reduces hallucination in large language models. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 3563–3578, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

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. *Preprint*, arXiv:2407.21783.

Felix Faltings, Michel Galley, Gerold Hintz, Chris Brockett, Chris Quirk, Jianfeng Gao, and Bill Dolan. 2021. Text editing by command. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5259–5274, Online. Association for Computational Linguistics.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. ELI5: Long form question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3558–3567, Florence, Italy. Association for Computational Linguistics.

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

Zhibin Gou, Zhihong Shao, Yeyun Gong, Yujiu Yang, Nan Duan, Weizhu Chen, et al. 2024. Critic: Large language models can self-correct with tool-interactive critiquing. In *The Twelfth International Conference on Learning Representations*.

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

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. 2024. Large language models cannot self-correct reasoning yet. In *The Twelfth International Conference on Learning Representations*.

Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong Park. 2024. Adaptive-RAG: Learning to adapt retrieval-augmented large language models through question complexity. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7036–7050, Mexico City, Mexico. Association for Computational Linguistics.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023a. Mistral 7B. *Preprint*, arXiv:2310.06825.

Che Jiang, Biqing Qi, Xiangyu Hong, Dayuan Fu, Yang Cheng, Fandong Meng, Mo Yu, Bowen Zhou, and Jie Zhou. 2024. On large language models' hallucination with regard to known facts. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1041–1053, Mexico City, Mexico. Association for Computational Linguistics.

Zhengbao Jiang, Frank Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023b. Active retrieval augmented generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7969–7992, Singapore. Association for Computational Linguistics.

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

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.

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

Patrick Lewis, Yuxiang Wu, Linqing Liu, Pasquale Minervini, Heinrich Küttler, Aleksandra Piktus, Pontus Stenetorp, and Sebastian Riedel. 2021. PAQ: 65 million probably-asked questions and what you can do with them. *Transactions of the Association for Computational Linguistics*, 9:1098–1115.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

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. *Preprint*, arXiv:1907.11692.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. *Proceedings of the International Conference on Learning Representations (ICLR)*, arXiv:1711.05101.

Hongyin Luo, Yung-Sung Chuang, Yuan Gong, Tianhua Zhang, Yoon Kim, Xixin Wu, Danny Fox, Helen Meng, and James Glass. 2023. SAIL: Search-augmented instruction learning. *Preprint*, arXiv:2305.15225.

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

Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. 2024. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12076–12100, Singapore. Association for Computational Linguistics.

Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2020. AmbigQA: Answering ambiguous open-domain questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5783–5797, Online. Association for Computational Linguistics.

Jianmo Ni, Gustavo Hernandez Abrego, Noah Constant, Ji Ma, Keith Hall, Daniel Cer, and Yinfei Yang. 2022. Sentence-T5: Scalable sentence encoders from pre-trained text-to-text models. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1864–1874, Dublin, Ireland. Association for Computational Linguistics.

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

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

Alec Radford. 2018. Improving language understanding by generative pre-training. Technical report, OpenAI.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2023. Exploring the limits of transfer learning with a unified text-to-text transformer. *Preprint*, arXiv:1910.10683.

Vipula Rawte, Swagata Chakraborty, Agnibh Pathak, Anubhav Sarkar, S.M Towhidul Islam Tonmoy, Aman Chadha, Amit Sheth, and Amitava Das. 2023. The troubling emergence of hallucination in large language models - an extensive definition, quantification, and prescriptive remediations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2541–2573, Singapore. Association for Computational Linguistics.

Tal Schuster, Adam Lelkes, Haitian Sun, Jai Gupta, Jonathan Berant, William Cohen, and Donald Metzler. 2024. SEMQA: Semi-extractive multi-source question answering. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1363–1381.

Ivan Stelmakh, Yi Luan, Bhuwan Dhingra, and Ming-Wei Chang. 2022. ASQA: Factoid questions meet long-form answers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8273–8288, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, New Orleans, Louisiana. Association for Computational Linguistics.

Gladys Tyen, Hassan Mansoor, Victor Carbune, Peter Chen, and Tony Mak. 2024. LLMs cannot find reasoning errors, but can correct them given the error location. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 13894–13908, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

Tianlu Wang, Ping Yu, Xiaoqing Ellen Tan, Sean O'Brien, Ramakanth Pasunuru, Jane Dwivedi-Yu, Olga Golovneva, Luke Zettlemoyer, Maryam Fazel-Zarandi, and Asli Celikyilmaz. 2023. Shepherd: A critic for language model generation. *arXiv preprint arXiv:2308.04592*.

Sean Welleck, Ximing Lu, Peter West, Faeze Brahman, Tianxiao Shen, Daniel Khashabi, and Yejin Choi. 2023. Generating sequences by learning to self-correct. In *The Eleventh International Conference on Learning Representations*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N Bennett, Junaid Ahmed, and Arnold Overwijk. 2020. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *International Conference on Learning Representations*.

Ziwei Xu, Sanjay Jain, and Mohan Kankanhalli. 2024. Hallucination is inevitable: An innate limitation of large language models. *arXiv preprint arXiv:2401.11817*.

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

Michihiro Yasunaga, Jure Leskovec, and Percy Liang. 2021. LM-critic: Language models for unsupervised grammatical error correction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7752–7763, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Muru Zhang, Ofir Press, William Merrill, Alisa Liu, and Noah A. Smith. 2024. How language model hallucinations can snowball. In *Forty-first International Conference on Machine Learning*.

A Appendix

A.1 Implementation Details

Models In our experiments, we employ two open-source LLMs Mistral 7B (Jiang et al., 2023a) and LLaMA-3.1 8B (Dubey et al., 2024) via Hugging Face (Wolf et al., 2020) API and GPT-4o (Achiam et al., 2023) which is accessible via OpenAI API, representing relatively small, medium, and large models, respectively. Here, these models are never fine-tuned or further trained except for their roles in verification. For the overall Streaming-VR pipeline, LLaMA-3.1 8B functions as the primary backbone language model to generate answers for given queries, while all three models are employed for verification or refinement for experiments.

Streaming Verifier We finetune Mistral 7B and LLaMA-3.1 8B as verifiers with the augmented training data derived from both ASQA and QuoteSum datasets. To elaborate, the verifier is trained for 5 epochs for each train set with the learning rate of 1e-5 and the AdamW (Loshchilov and Hutter, 2019) as its optimizer.The final test results of the finetuned verifiers used in the experiments, including a Random baseline that selects verification results arbitrarily, are presented in Table 5. For the entire pipeline, we establish a constraint that prohibits the use of any other models larger than the backbone as verifiers. This decision is based on the principle that the verifier should not exceed the capabilities of the backbone model, as the verifier serves merely as a supplementary tool for identifying mistakes. This reflects our considerations regarding computational costs and efficiency.

Table 5: The results of Streaming Verifier finetuned on train set for each dataset. We report the test accuracy of verifiers along with random classifier.

Method	ASQA	QuoteSum
Random	49.6	50.3
Mistral 7B	86.8	81.7
LLaMA-3.1 8B	86.7	93.0

Retrieval Augmented Generation To evaluate the consistent impact of Streaming-VR also in a Retrieval-Augmented Generation (RAG) setting, as in the original ASQA paper (Stelmakh et al., 2022), we perform experiments using retrieved documents. Specifically, we use the top-k documents ranked by semantic similarity between the query and external documents for open-book answer generation on the ASQA dataset. These documents, retrieved from the Wikipedia corpus (2018-12-20 snapshot) using GTR-XXL (Ni et al., 2021), are provided by the LLM citation benchmark ALCE (Gao et al., 2023).

In-context Learning Building on the original evaluation of QuoteSum (Schuster et al., 2024), we further conduct a quantitative assessment of the variants of few-shot backbone models. Specifically, we use dynamic prompt with top-k examples for each questions in the test set. These examples are retrieved from the training set by selecting the passages whose queries are most similar to the target test query, based on the cosine similarity between their sentence embeddings (Ni et al., 2022).