Towards Verifiable Text Generation with Evolving Memory and Self-Reflection

Anonymous ACL submission

Abstract

Despite the remarkable ability of large language models (LLMs) in language comprehension and generation, they often suffer from producing factually incorrect information, also known as hallucination. A promising solution to this issue is verifiable text generation, which prompts LLMs to generate content with citations for accuracy verification. However, verifiable text generation is non-trivial due to the focus-shifting phenomenon, the intricate reasoning needed to align the claim with correct citations, and the dilemma between the precision and breadth of retrieved documents. In this paper, we present **VTG**, an innovative framework for **V**erifiable **T**ext **G**eneration with evolving memory and self-reflection. VTG introduces evolving long short-term memory to retain both valuable documents and recent documents. A two-tier verifier equipped with an evidence finder is proposed to rethink and reflect on the relationship between the claim and citations. Furthermore, active retrieval and diverse query generation are utilized to enhance both the precision and breadth of the retrieved documents. We conduct extensive experiments on five datasets across three knowledge-intensive tasks and the results reveal that VTG significantly outperforms baselines.

1 Introduction

Large Language Models (LLMs) (Scao et al., 2022; Taylor et al., 2022; Chowdhery et al., 2022) have showcased remarkable performance across a spectrum of downstream tasks recently. Despite their advancements, LLMs often generate responses that include hallucinated facts and inaccurate information (Ji et al., 2023; Shuster et al., 2021; Zhang et al., 2023a), undermining their reliability.

To enhance the reliability of LLMs, a new generation paradigm, Verifiable Text Generation (Gao et al., 2023b, 2022; Bohnet et al., 2022; Liu et al., 2023a; Li et al., 2023a; Funkquist et al., 2022),

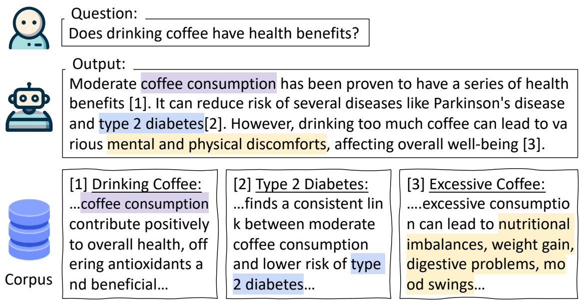


Figure 1: Given a question, the system generates text while providing citing documents from a large corpus.

is proposed to encourage LLMs to provide citations for any claim they generate. For example, as shown in Figure 1, the response to the question "Does drinking coffee have health benefits?" contains authentic sources supporting the claims, enhancing its credibility. In this way, verifiable generation produces more trustworthy answers, which facilitates its application in multiple commercial systems, such as Bing Chat[1] and perplexity.ai[2].

However, verifiable text generation is challenging for the following reasons. **Firstly**, it often involves long text generation, where the focus of the content changes over time, characterized by focus-shifting phenomenon (Lan and Jiang, 2021; Sun et al., 2023a). This dynamic poses challenges in consistently aligning claims with the appropriate evidential references. For instance, as depicted in Figure 1, the discussion evolves from the health benefits of moderate coffee consumption to the adverse effects of excessive consumption. Such shifts demand a dynamic adaptation in the document pool to support the shifting focus of the generation. **Secondly**, identifying the intricate relationship between a claim and its potential evidence requires more than just linguistic matching—it demands careful analysis. For example, in Figure 1, the

[1]https://www.bing.com/new
[2]https://www.perplexity.ai

third claim suggests excessive coffee consumption leads to various mental and physical discomforts. The corresponding supporting document, although not explicitly mentioning these discomforts, describes symptoms inherently related to them. This necessitates an in-depth examination to confirm that the evidence truly supports the specific claim. **Thirdly**, striking a balance between the precision and breadth of retrieved documents presents a complex challenge for verifiable text generation. On the one hand, the task is susceptible to noisy documents during the claim-citation alignment process, emphasizing the need to selectively retain a few highly relevant documents. On the other hand, the intrinsic nature of verifiable text generation calls for a comprehensive collection of documents to enhance credibility. Therefore, crafting strategies to balance precision and breadth in document retrieval is crucial for advancing verifiable text generation.

When composing text with citations, individuals are capable of adaptively gathering the most relevant information regarding the claim being written, typically involving active information seeking and frequent verification. Inspired by this process, we propose **VTG**, short for **V**erifiable **T**ext **G**eneration, a novel framework that operates through iterative generation and verification, utilizing an evolving memory and a two-tier verifier. Specifically, to address the challenge of focus-shifting, VTG employs an evolving long short-term memory system. This system effectively archives important documents in long-term memory and maintains recent ones in short-term memory, thereby providing support for the evolving focus of the generation. Moreover, to identify the complex relationship between a claim and its potential evidence, VTG employs a generation verifier and a memory verifier, both using Natural Language Inference (NLI) model to assess the logical support of potential evidence for the claim. The generation verifier first checks if the cited documents logically support the claim. If there's a misalignment, the memory verifier reevaluates the claim against the documents stored in memory. A positive outcome suggests that the misalignment is due to the citation generation process, not because the information in the claim is wrong, leading to the adoption of a refined set of documents from memory for citation. Conversely, a negative outcome indicates potential factual inaccuracies in the claim, triggering an evidence finder to gather external information, which facilitates the regeneration of a more accurate and verifiable claim. Lastly, to balance between precision and breadth in document retrieval, VTG incorporates active retrieval and diverse query generation. Retrieval is initiated only when the claim does not pass the memory verifier, indicating potential factual inaccuracies. This approach guarantees the necessity of retrieval, reducing noise from unnecessary retrieval, and thereby enhancing retrieval precision. By instructing LLMs to generate diverse queries, the breadth of retrieved documents is broadened, enabling the documents to offer comprehensive support for the claim.

To summarize, our main contributions are:

- We introduce VTG, a novel framework that guides the generation model using the combination of an evolving memory and a two-tier verifier, offering an adaptive and reflective approach for verifiable text generation.
- The evolving memory stores valuable and recent documents, effectively addressing the focus-shifting challenge. The two-tier verifier and evidence finder enable the in-depth examination of the claim and its potential evidence. The active retrieval and diverse query generation can improve both the precision and breadth of the retrieved documents.
- We conduct extensive experiments on five datasets across three knowledge-intensive tasks and the results show that VTG significantly outperforms baselines on both citation quality and answer correctness.

2 Methodology

In this section, we first present the overall framework of VTG. Then we will go over each part of the model in detail.

2.1 Overall Framework

Given a question q and a corpus of text passages $\mathcal{D}$, the task of verifiable text generation demands the system to return an output $\mathcal{S}$, which consists of n claims, and each claim s_i cites a list of passages $\mathcal{C}_i = \{c_{i,1}, c_{i,2}, \ldots\}$. As shown in Figure 2, VTG operates with an evolving memory system: the long-term memory D_L that is maintained throughout the generation process, and the short-term memory D_S that is continually updated to align with the shifting focus of content. Initially, D_L is filled with the top-k retrieved documents based on the original question, while D_S starts empty. The LLM generates the first claim and its corresponding citation

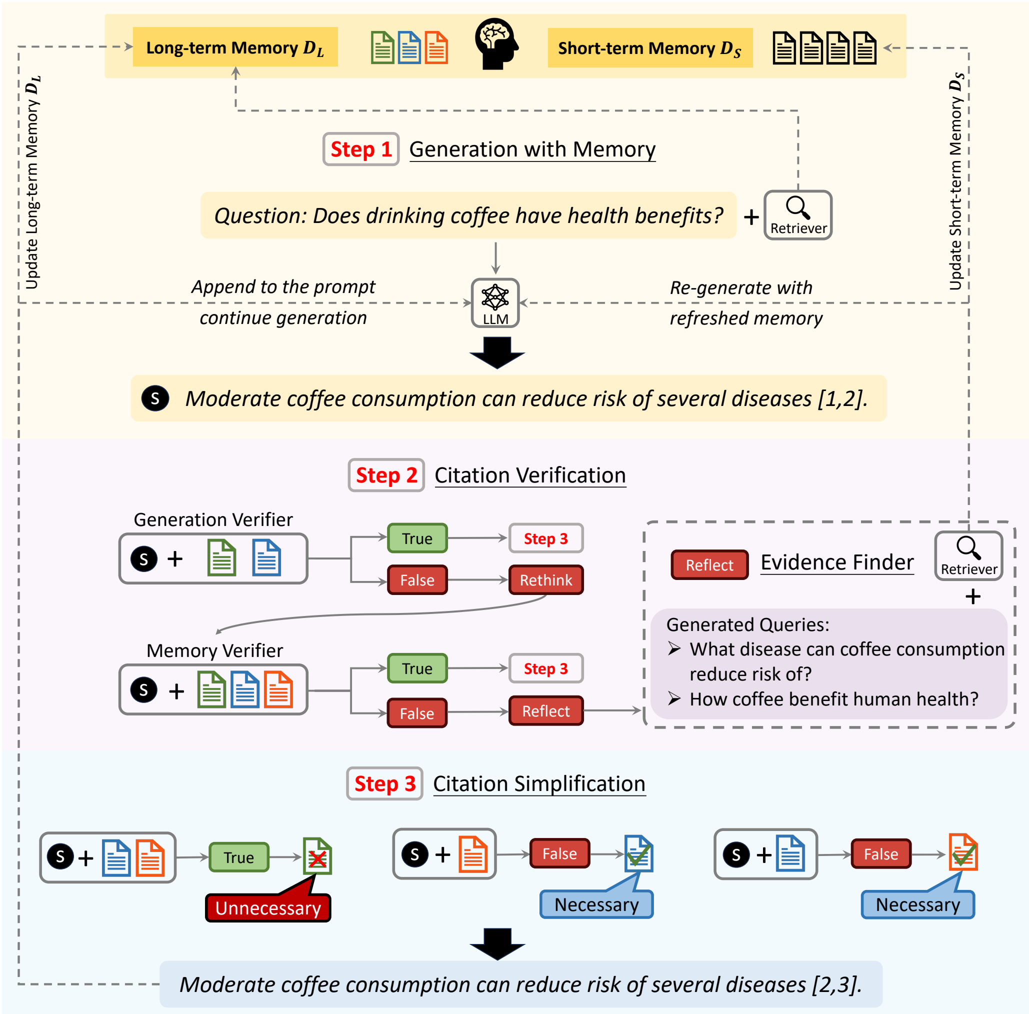


Figure 2: The illustration of VTG, which mainly consists of three stages: Generation with Memory, Citation Verification, and Citation Simplification. The Evidence Finder will only be activated when the claim fails to pass the memory verifier, indicating potential factual inaccuracies in the claim.

based on documents from both memories. Subsequently, the generation verifier examines whether the cited documents logically support the claim. If the outcome is negative, the memory verifier then evaluates whether the combined memories (i.e., $D_S \cup D_L$) logically support the claim. If either the cited documents or the documents in the memories support the claim, the corresponding document set will undergo simplification and be incorporated into D_L for future generations. Conversely, if neither of them supports the claim, the LLM generates diverse queries about the claim, which are then used to retrieve documents to refresh D_S. In cases where a regenerated claim fails to pass verification after T attempts, the model progresses to the next stage of generation. This interactive and iterative process continues until the LLM generates an end token, indicating generation completion.

2.2 Generation with Evolving Memory

In verifiable text generation, the content's focus may shift as the generation progresses, known as the focus-shifting phenomenon (Lan and Jiang, 2021; Sun et al., 2023a). Consequently, the document pool must be dynamically adjusted to ensure that the documents can support the latest claim. This variability presents a challenge when generating claims and citations simultaneously due to fluctuating reference indices in previously generated content, which could potentially bring confusion into the model's generation process. Moreover, insights from previous research (Juneja et al., 2023) suggest that tasks requiring distinct skills are more effectively handled through separate modules instead of one monolithic module. Based on these insights, VTG adopts a divided approach: dividing the verifiable text generation task into distinct claim generation and citation generation processes.

Claim Generator aims to complete unfinished content based on the documents from both long-term memory D_L and short-term memory D_S. Initially, long-term memory D_L is filled with top-k documents selected based on their relevance to the original question. This ensures that information closely related to the original question is permanently retained in memory, providing a consistent guide throughout the entire generation process. In contrast, short-term memory D_S starts as an empty set, ready to be dynamically filled as needed.

Citation Generator aims to source supporting evidence for the generated claim from both long-term memory D_L and short-term memory D_S. This focused approach allows the citation generator to concentrate on validating the current claim without being distracted by unrelated contexts.

2.3 Citation Construction

Model hallucination in LLM outputs often leads to inaccuracies and distortions (Ji et al., 2023; Shuster et al., 2021; Zhang et al., 2023a). One approach to mitigate this issue is by grounding the model's outputs in specific documents. However, even citations generated by LLM can be subject to hallucinations, potentially leading to unreliable or irrelevant references. To ensure that the evidence truly supports the specific claim, VTG employs a two-tier verification system comprising the generation verifier and the memory verifier, both of which use Natural Language Inference (NLI) model to assess the logical support of potential evidence (premise) for the claim (hypothesis). The generation verifier assesses whether the cited documents logically support the claim, while the memory verifier evaluates the support provided by the entire memory set. Upon successful verification by either component, the citation set advances to the citation simplifier to remove any unnecessary references.

Generation Verifier examines whether the citation set logically supports the claim. If the verification is successful, the citation set advances to the citation simplifier for further refinement. However, if the verification fails, there are two possible reasons: 1) The claim is logically consistent with the memory set, but the generated citation set is inaccurate. In this case, the citation simplifier removes unnecessary documents from the full memory set, and the remaining documents are used as the citation. 2) The claim lacks support from the memory set, suggesting potential factual inaccuracies in the claim. Here, the evidence finder is activated to seek relevant information for claim regeneration.

Memory Verifier detects the potential hallucination of the current claim by analyzing whether the full memory set logically supports it. A positive outcome suggests that the claim-citation misalignment originates from the citation generation process, triggering the citation simplifier to refine the full memory set to be used as the final citation. However, if the full memory set still cannot validate the current claim, it implies that the LLM may have fabricated the claim with its parametric knowledge and the claim might be factually incorrect, necessitating the employment of the evidence finder to seek evidence for claim regeneration. By doing so, the proposed framework is able to assess the generated citations in a self-reflection manner.

Citation Simplifier is designed to eliminate unnecessary references. It works by iteratively reviewing each document in the citation set, temporarily removing one, and assessing if the claim remains well-supported without it. Redundant citations, that do not contribute to the claim's verification, are permanently removed. This iterative process ensures the final citation set is concise and supportive, retaining only essential citations for claim verification. These citations, after verification and simplification, become part of long-term memory D_L to guide future claim generations.

2.4 Evidence Finder

When the full memory set fails to pass the memory verifier, indicating potential factual inaccuracies in the generated claim, the evidence finder is activated to retrieve relevant documents for verification.

The intrinsic nature of verifiable text generation necessitates a wide range of documents to boost the credibility of the generated content. To expand the knowledge scope of the retrieved documents, VTG prompt LLMs to formulate queries that explore various aspects of the current claim. Furthermore, traditional query generation methods, which typically rely solely on the current claim, can lead to ambiguity, particularly with claims containing pronouns or unclear references. To overcome this, VTG introduces a context-aware query generation approach. This method enhances query generation by incorporating the original question, the current claim and the unfinished content into the prompt.

Once queries are generated, a retriever collects relevant documents for each query. These documents then update the short-term memory D_S, with the latest and most relevant information. This key update ensures that the short-term memory is both current and pertinent, thereby enabling LLM to provide precise citations for the latest claim.

3 Experiment

3.1 Baselines

For an equitable comparison, we have selected the following four best-performing baseline methodologies as proposed in ALCE (Gao et al., 2023b).

VANILLA: The LLM generates responses with citations based on the top-ranked documents.

SUMM: The LLM first summarizes information from the top-ranked documents and then generates texts with citations based on the summarization.

SNIPPET: The LLM first extracts relevant snippets from the top-ranked documents and then generates texts with citations based on the snippets.

RERANK: The LLM first generates four unique responses using high temperature and outputs the one with the highest citation recall.

Besides, we also compare with several post-processing baselines, which include **POSTCITE**, **REFINECITE**, **VERICITE** and **VERIREFINE**. Due to space limits, we only put the four baselines from ALCE in the main results, for the complete experiment results, please refer to Appendix A.

3.2 Datasets and Evaluation

We assess the effectiveness of our methods on five datasets across three knowledge-intensive tasks. For all datasets, our evaluation criteria encompass both the answer correctness and citation quality of model outputs. The details of the tasks and the datasets we used are as follows:

Multihop QA entails answering complex questions that necessitate multiple retrieval and reasoning steps (Yang et al., 2018; Ho et al., 2020). We employ the 2WikiMultihopQA dataset (Ho et al., 2020), which consists of 2-hop complex questions derived from Wikipedia, requiring skills in composition, comparison or inference.

In line with Jiang et al. (2023), LLMs are prompted to provide the final answer, which is then evaluated against the reference answer using answer-level Exact Match (EM), token-level precision, recall and F1 metrics.

Long-form QA aims to generate detailed answers to complex questions (Fan et al., 2019a,b), we choose the ASQA dataset (Stelmakh et al., 2022) and the ELI5 dataset (Fan et al., 2019b) for evaluation. ASQA focuses on ambiguous questions requiring comprehensive answers covering multiple interpretations. ELI5, on the other hand, deals with complex questions demanding lengthy, in-depth answers backed by multiple documents.

For ASQA, we apply the metrics outlined in Stelmakh et al. (2022), including Exact Match (EM), a soft match using a RoBERTa-based QA model (Disambig-F1), ROUGE (Lin, 2004) and a combined DR score. In the case of ELI5, we adhere to the evaluation criteria of Gao et al. (2023b), focusing on whether the model's predictions address the sub-claims of the gold-standard answer.

Open-domain QA requires leveraging external knowledge for answering questions. We choose the NQ dataset (Kwiatkowski et al., 2019) and the WebQ dataset (Berant et al., 2013) for evaluation.

Following the methodology of Yu et al. (2022) and Sun et al. (2023b), LLMs are prompted to generate the final answer, which is then compared with the reference answer using answer-level Exact Match (EM).

Verifiability Evaluation. To evaluate the citation quality of responses, we employ the approach of Gao et al. (2023b), focusing on calculating *ALCE.Citation Recall*, *ALCE.Citation Precision*, and the combined *ALCE.Citation F1* score. *ALCE.Citation Recall* examines whether the output is fully supported by the cited documents, while *ALCE.Citation Precision* assesses the redundancy of the citations included.

Additionally, to further improve the robustness of the evaluation, we incorporate the use of a LLM (Qwen-MAX) as the citation evaluator. Specifically, The large language model (LLM) is given a sentence and all the passages that the sentence cited and is asked to judge whether the passages fully support the sentence, which is used as the signal to compute *LLM.Citation Recall* and *LLM.Citation Precision*.

For more details on dataset statistics and evaluation details, please refer to Appendix B.

Datasets	Wikihop								WebQ							NQ						
	Correct		ALCE.Citation			LLM.Citation			Correct	ALCE.Citation			LLM.Citation			Correct	ALCE.Citation			LLM.Citation		
Metrics	EM	F1	Rec	Prec	F1	Rec	Prec	F1	EM	Rec	Prec	F1	Rec	Prec	F1	EM	Rec	Prec	F1	Rec	Prec	F1
											Vicuna-13B											
VANILLA	23.40	21.98	29.55	22.25	25.39	41.59	35.04	38.03	55.80	67.66	60.66	63.97	67.50	67.83	67.67	54.80	71.39	61.71	66.20	77.46	63.94	70.05
SUMM	23.20	20.00	30.89	28.43	29.61	37.66	39.12	38.37	58.00	70.51	62.07	66.02	68.23	66.57	67.39	57.00	51.55	52.21	51.88	56.76	62.86	59.66
SNIPPET	21.80	20.05	25.18	21.95	23.45	33.67	29.75	31.59	58.40	53.44	49.15	51.21	68.46	69.57	69.01	57.20	43.56	41.43	42.47	57.56	59.57	58.55
RERANK	22.60	21.13	47.03	47.53	47.28	54.33	53.73	54.03	56.40	89.93	76.33	82.57	88.20	67.77	76.64	56.20	83.56	73.57	78.25	81.66	73.49	77.36
VTG	**25.60**	**23.27**	**55.36**	**49.59**	**52.32**	**62.76**	**54.69**	**58.45**	**60.00**	**92.16**	**86.51**	**89.25**	**89.43**	**81.38**	**85.21**	**58.00**	**88.69**	**82.02**	**85.22**	**86.35**	**78.06**	**82.00**
											Text-Davinci-003											
VANILLA	33.00	33.01	40.46	28.30	33.30	59.07	43.00	49.77	67.50	63.78	58.97	61.28	71.18	66.52	68.77	62.50	60.48	55.56	57.92	66.45	61.59	63.93
SUMM	30.00	30.63	9.39	12.19	10.61	23.19	24.64	23.89	67.50	60.06	47.62	53.12	68.33	56.65	61.95	62.50	44.23	38.45	41.14	55.36	49.20	52.10
SNIPPET	32.00	30.13	13.86	18.49	15.84	37.36	38.99	38.16	67.00	65.41	52.32	58.14	71.81	68.07	69.89	62.00	54.72	46.99	50.56	73.05	69.55	71.25
RERANK	32.67	33.09	56.13	45.22	50.09	63.43	46.32	53.54	67.00	73.72	64.90	69.03	78.12	70.00	73.84	61.50	71.30	63.44	67.14	79.03	66.37	72.15
VTG	**41.50**	**40.19**	**63.89**	**57.65**	**60.61**	**70.47**	**59.13**	**64.30**	**68.00**	**93.00**	**88.72**	**90.81**	**90.70**	**87.52**	**89.08**	**63.00**	**91.85**	**86.59**	**89.14**	**84.92**	**73.20**	**78.63**

Table 1: Comparisons between VTG and baselines on Multi-hop QA task and Open-domain QA task.

Datasets	ASQA										ELI5							Overall	
	Correct				ALCE.Citation			LLM.Citation			Correct	ALCE.Citation			LLM.Citation			Correct	Citation
Metrics	EM	D-F1	R-L	DR	Rec	Prec	F1	Rec	Prec	F1	Claim	Rec	Prec	F1	Rec	Prec	F1	EM	F1
										Vicuna-13B									
VANILLA	32.00	27.52	33.53	30.53	72.78	62.09	67.01	73.28	66.59	69.78	12.20	59.79	48.26	53.41	81.46	76.89	79.11	35.64	60.06
SUMM	41.71	28.95	37.18	33.07	62.15	59.60	60.85	68.95	70.19	69.56	14.20	60.13	52.42	56.01	77.87	72.42	75.05	38.82	57.44
SNIPPET	39.22	27.01	35.65	31.33	46.23	47.04	46.63	56.55	63.03	59.61	14.33	31.47	32.72	32.08	46.69	49.32	47.97	38.19	46.26
RERANK	37.14	28.21	32.18	30.20	88.29	75.74	81.53	88.29	75.74	81.53	11.67	73.80	61.12	66.86	84.57	77.09	80.65	36.80	72.67
VTG	**41.92**	**30.53**	**37.87**	**34.20**	**89.15**	**82.57**	**85.73**	**89.15**	**82.57**	**85.73**	**14.73**	**81.50**	**72.16**	**76.55**	**87.60**	**84.46**	**86.00**	**40.05**	**78.65**
										Text-Davinci-003									
VANILLA	40.25	31.47	35.81	33.64	58.13	55.17	56.61	58.13	55.17	56.61	13.43	58.66	47.40	52.43	58.66	47.40	52.43	43.34	55.31
SUMM	41.33	28.91	37.21	33.06	48.31	40.68	44.17	50.48	44.44	47.27	11.50	39.43	31.81	35.21	52.27	48.47	50.30	42.57	41.98
SNIPPET	39.60	30.11	38.35	34.23	53.14	43.19	47.65	59.31	52.05	55.44	13.67	45.29	37.23	40.87	62.39	55.19	58.57	42.85	50.64
RERANK	39.55	29.94	39.38	34.66	75.83	69.81	72.70	76.41	70.01	73.07	14.76	76.21	61.67	68.17	86.98	77.64	82.04	43.10	68.18
VTG	**41.53**	**31.64**	**39.45**	**35.55**	**86.70**	**79.95**	**83.19**	**89.10**	**79.84**	**84.22**	**16.67**	**82.63**	**71.56**	**76.70**	**87.94**	**81.79**	**84.75**	**46.14**	**80.14**

Table 2: Comparisons between VTG and baselines on Long-form QA task and overall performance.

3.3 Implementation Details

To prove the generalizability of our method, we conduct experiments using LLMs of different parameter sizes. Specifically, we utilize two LLMs: `Vicuna-13B-v1.5-16k`[3] (Zheng et al., 2023) and `Text-Davinci-003`[4] (Ouyang et al., 2022) for evaluation, respectively. For the evaluation of verifiability and the inference tasks in both the RERANK and VTG methods, we employ the TRUE[5] model (Raffel et al., 2020), a T5-11B model fine-tuned on a collection of NLI datasets, to automatically examine whether the cited documents entail the claim. Following Gao et al. (2023b), we use Wikipedia dump from Dec. 20, 2018 as our retrieval corpus and use DPR (Karpukhin et al., 2020) as our dense retriever.

3.4 Main Results

In this section, we present a comparison of the performance of VTG against other baselines across five different datasets in Tables 1 and 2. Based on these results, several observations can be made:

First, our proposed VTG consistently outperforms other approaches across various datasets and metrics when applied to LLMs of different parameter sizes. Notably, VTG achieves a significant enhancement in citation quality, with a notable 22% and 9% relative improvement over the strongest competitor RERANK, when evaluated with `Text-Davinci-003` and `Vicuna-13B`, respectively. Moreover, VTG's strong capability for verifiable generation also leads to a considerable improvement in answer correctness, evidenced by an approximate 5% overall improvement compared to the leading baselines across both LLMs.

Second, among the evaluated baselines, the method RERANK stands out in the aspect of citation quality, primarily owing to its multiple sampling strategy that enhances the chances of producing high-quality outputs. However, its performance in answer correctness fluctuates across different datasets, which is notably lower in the ASQA dataset when evaluated with `Text-Davinci-003` and in the ELI5 dataset when evaluated with `Vicuna-13B`. In contrast, VTG demonstrates consistent improvement across all metrics and datasets, highlighting its robustness and reliability.

Third, the comparison between the two different LLMs reveals interesting findings. By integrating a broader range of documents, SUMM and SNIPPET outperform VANILLA in terms of overall cor-

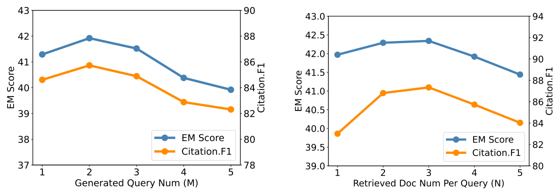
[3] `https://huggingface.co/lmsys/vicuna-13b-v1.5-16k`
[4] `https://api.openai.com/v1/completions` as of October 2023
[5] `https://huggingface.co/google/t5_xxl_true_nli_mixture`

	Correct		Citation		
	EM	F1	Rec	Pre	F1
VTG	**25.60**	**23.27**	**55.36**	**49.59**	**52.32**
-w/o Verifier	20.60	17.75	37.34	30.16	33.36
-w/o Memory	21.40	18.88	43.88	35.18	39.05
-w/o Simplifier	24.20	22.85	45.07	28.61	36.00
-w/o Diverse QG	21.40	18.99	47.76	40.15	43.62

Table 3: Ablation Study on 2WikiMultihopQA.

	Correct		Citation			Cost
T	EM	F1	Rec	Prec	F1	Trials
1	22.00	19.56	41.20	34.07	37.29	1.82
2	22.60	20.22	47.82	41.95	44.69	2.15
3	22.80	20.63	49.95	44.03	46.80	2.46
4	25.00	22.66	51.37	45.70	48.36	2.72
5	25.60	23.27	55.36	49.59	52.31	2.92

Table 4: Performance of VTG with respect to the max trials T on 2WikiMultihopQA, where the "Trials" represent the average iteration it takes to complete a claim.



(a) Quey Generation Num (b) Retrieval Num

Figure 3: The performance change over different hyperparameters on ASQA.

rectness when evaluated with `Vicuna-13B`. However, this advantage diminishes when evaluated with `Text-Davinci-003`. This could be attributed to `Text-Davinci-003`'s extensive internal knowledge base, which enables it to generate answers without relying on external sources, as evidenced by the performance comparison when applying the same method to both LLMs. Consequently, SUMM and SNIPPET may introduce unnecessary noisy information in the context of `Text-Davinci-003`, leading to correctness degradation. Additionally, these methods struggle with citation quality on both LLMs, as simplified documents make it hard for LLMs to generate the correct citations.

3.5 Analysis

In this section, we conduct analytical experiments of our method using `Vicuna-13B` as the default LLM, unless specified otherwise.

Ablation Study. We assess the impact of each component in our model using the 2WikiMultihopQA dataset. By removing components one by one, we observe their individual contributions to performance. "w/o Verifier" excludes the two-tier verifier during generation, "w/o Memory" removes the evolving memory system, "w/o Simplifier" removes the citation simplifier and "w/o Diverse QG" replaces diverse query generation with direct retrieval using the claim as the query.

Results in Table 3 show that removing any component decreases performance, highlighting their importance. Notably, the removal of the verifier results in the most significant drop in performance, as it potentially leads to the generation of claims with hallucinations or factual inaccuracies. Omitting the simplifier has a less pronounced effect on correctness, as all claims are still verified. However, it does lead to unnecessary citations, which reduces the precision of citation quality. Removing the memory component also results in a decline in performance, affecting both correctness and citation quality. This is primarily due to the lack of supporting evidence for the constantly changing topic. Lastly, removing diverse QG limits the model's ability to retrieve a broader range of relevant documents, leading to a degradation in performance.

Performance over Max Trials. We examine the impact of various max trials T on VTG's performance using 2WikiMultihopQA dataset. As illustrated in Table 4, we observed that increasing the value of T correlates with improved performance in terms of correctness and citation quality. This improvement is reasonable since a higher T allows the model more attempts to generate a claim that passes the verification process, increasing the chances of generating accurate and well-supported claims. However, it's important to acknowledge that a higher T also leads to larger token consumption, indicating the need to adjust T to balance between effectiveness and computational cost.

Retrieval Analysis. We examine the impact of retrieval parameters on VTG's performance using ASQA dataset, focusing on the number of generated queries M and the number of retrieved documents per query N. As shown in Figure 3, we observe a consistent trend for both parameters. Initially, increasing M and N enhances both correctness and citation quality, which can be attributed to the fact that a larger pool of documents offers a

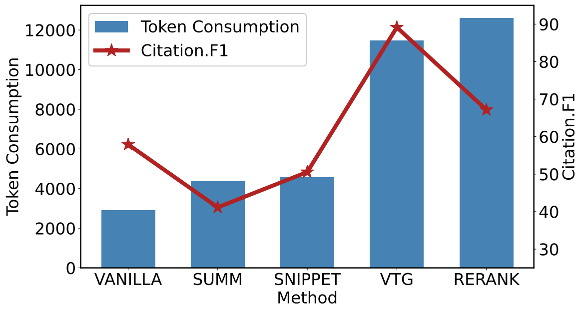


Figure 4: Token Consumption Comparison on NQ.

broader knowledge scope and provides more citation options for the LLM. However, continually increasing them beyond a certain threshold results in a decline in performance, which is mainly because an excessively large document pool can introduce noisy information, negatively impacting both claim generation and citation generation processes.

Token Consumption Analysis. We analyze token consumption across various methods on the NQ dataset using `Text-Davinci-003` as the LLM. As shown in Figure 4, VANILLA achieves the lowest token cost due to its single API request. After introducing more documents, SUMM, SNIPPET incur higher token costs but with lower citation quality as simplified documents challenge LLMs' citation generation. RERANK produces better citation quality than other baselines but incurs the highest token cost due to its strategy of multiple sampling. In contrast, our method VTG demonstrates a lower token cost than RERANK while significantly improving citation quality, highlighting the superiority of our approach. Importantly, users can adjust VTG's token cost with the max trial parameter T to balance performance and computational cost.

4 Related Work

4.1 Retrieval-augmented LLMs.

Retrieval-augmented LLMs aim to provide extra documents to the LLM, which has been proven useful in many knowledge-intensive tasks. Among the existing studies, Shi et al. (2023); Wang et al. (2023b); Zhang et al. (2023b); Yu et al. (2023a,c) propose to retrieve only once at the beginning. Other works (Qian et al., 2023; Yu et al., 2023b) propose to retrieve multiple times during generation, which offers the flexibility of when and what to search. For example, Jiang et al. (2023) propose to retrieve when the generation contains low confidence tokens. Ram et al. (2023) propose to refresh the retrieved document every n token, which is demonstrated to be more effective than retrieving only once. Wang et al. (2023a); Asai et al. (2023); Zhao et al. (2023) propose to retrieve only when the LLMs need to. Among the existing studies, retrieve on-the-fly methods are the closest to ours. However, these methods do not provide referenced documents for the generated sentences, potentially reducing the reliability of the generated content.

4.2 Verifiable Text Generation

Verifiable Text Generation aims to generate content with supporting documents, which has been attracting attention in recent years. For example, Liu et al. (2023b); Qin et al. (2023); Nakano et al. focus on training LLMs to browse web pages and answer questions with evidence. Gao et al. (2023a) introduced the research-and-revision (RARR) method for retrieving evidence for LLM outputs. Li et al. (2023b) incorporated knowledge graphs as an evidence source. Other works mainly focus on evaluation (Liu et al., 2023a). For example, Rashkin et al. (2023) propose *Attributable to Identified Sources* (AIS) for human evaluation. Gao et al. (2022) define auto-AIS to approximate human AIS judgments. Gao et al. (2023b) propose ALCE to automate the evaluation of the citation quality. Min et al. (2023) proposed FactScore for evaluating the verifiability of generated facts. Although these methods achieve promising results, they do not effectively address the focus-shifting phenomenon and fail to capture the complex relationship between claims and citations. In this paper, we propose to maintain an evolving memory and two-tier verification to deal with the above-mentioned issues.

5 Conclusion

In this paper, we introduce VTG, a novel framework tailored to address the challenges of verifiable text generation. Central to its design is an evolving long short-term memory, which adaptively keeps both valuable documents and up-to-date documents. The two-tier verifier coupled with an evidence finder facilitates a deeper analysis and reflection on the relationship between claims and citations. Through the integration of active retrieval mechanisms and diverse query generation, VTG skillfully enhances both the precision and breadth of the document retrieval process. Extensive experiments on five datasets across three knowledge-intensive tasks verify the effectiveness of our method.

Limitations

In this work, we propose a novel frame VTG for verifiable text generation. The limitations of the proposed method are as follows: (1) the computational cost of VTG is relatively high due to the need for multiple API calls and frequent verification. This may restrict its applicability in resource-intensive scenarios or systems with limited computational resources; (2) the effectiveness of our verification process is constrained by the precision of the NLI models. In instances where the NLI model's accuracy is suboptimal, there is a risk of incorporating erroneous information into the method, potentially compromising the verifiability of the generated text.

As for future work, we plan to mitigate the computational cost of the method by developing more efficient pipelines. Moreover, we aim to reduce our approach's reliance on NLI models, thereby enhancing the overall robustness of our framework.

Ethics Statement

This work was conducted in rigorous compliance with the ACL Ethics Policy. All datasets and large language models (LLMs) used for evaluation are publicly available. Furthermore, our work strives to improve the verifiability of LLM outputs, which could potentially broaden the application scenarios of LLMs. We do not foresee any form of negative ethical impact induced by our work.

References

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*.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on freebase from question-answer pairs. In *EMNLP 2013*, pages 1533–1544.

Bernd Bohnet, Vinh Q Tran, Pat Verga, Roee Aharoni, Daniel Andor, Livio Baldini Soares, Jacob Eisenstein, Kuzman Ganchev, Jonathan Herzig, Kai Hui, et al. 2022. Attributed question answering: Evaluation and modeling for attributed large language models. *arXiv preprint arXiv:2212.08037*.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019a. ELI5: long form question answering. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 3558–3567. Association for Computational Linguistics.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019b. ELI5: Long form question answering. In *Association for Computational Linguistics (ACL)*, pages 3558–3567.

Martin Funkquist, Ilia Kuznetsov, Yufang Hou, and Iryna Gurevych. 2022. Citebench: A benchmark for scientific citation text generation. *arXiv preprint arXiv:2212.09577*.

Luyu Gao, Zhuyun Dai, Panupong Pasupat, Anthony Chen, Arun Tejasvi Chaganty, Yicheng Fan, Vincent Zhao, Ni Lao, Hongrae Lee, Da-Cheng Juan, et al. 2023a. Rarr: Researching and revising what language models say, using language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16477–16508.

Luyu Gao, Zhuyun Dai, Panupong Pasupat, Anthony Chen, Arun Tejasvi Chaganty, Yicheng Fan, Vincent Y Zhao, Ni Lao, Hongrae Lee, Da-Cheng Juan, et al. 2022. Rarr: Researching and revising what language models say, using language models. *arXiv preprint arXiv:2210.08726*.

Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. 2023b. Enabling large language models to generate text with citations. *arXiv preprint arXiv:2305.14627*.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing A multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pages 6609–6625. International Committee on Computational Linguistics.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

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

Gurusha Juneja, Subhabrata Dutta, Soumen Chakrabarti, Sunny Manchanda, and Tanmoy Chakraborty. 2023. Small language models fine-tuned to coordinate larger language models improve complex reasoning. *arXiv preprint arXiv:2310.18338*.

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, et al. 2019. Natural questions: A benchmark for question answering research. *TACL 2019*, pages 452–466.

Yunshi Lan and Jing Jiang. 2021. Modeling transitions of focal entities for conversational knowledge base question answering. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3288–3297.

Dongfang Li, Zetian Sun, Xinshuo Hu, Zhenyu Liu, Ziyang Chen, Baotian Hu, Aiguo Wu, and Min Zhang. 2023a. A survey of large language models attribution. *arXiv preprint arXiv:2311.03731*.

Xinze Li, Yixin Cao, Liangming Pan, Yubo Ma, and Aixin Sun. 2023b. Towards verifiable generation: A benchmark for knowledge-aware language model attribution. *arXiv preprint arXiv:2310.05634*.

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.

Nelson F Liu, Tianyi Zhang, and Percy Liang. 2023a. Evaluating verifiability in generative search engines. *arXiv preprint arXiv:2304.09848*.

Xiao Liu, Hanyu Lai, Hao Yu, Yifan Xu, Aohan Zeng, Zhengxiao Du, Peng Zhang, Yuxiao Dong, and Jie Tang. 2023b. Webglm: Towards an efficient web-enhanced question answering system with human preferences. *arXiv preprint arXiv:2306.07906*.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. Factscore: Fine-grained atomic evaluation of factual precision in long form text generation. *arXiv preprint arXiv:2305.14251*.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. Webgpt: browser-assisted question-answering with human feedback (2021). *URL https://arxiv.org/abs/2112.09332*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems (NeurIPS)*, 35:27730–27744.

Hongjin Qian, Zhicheng Dou, Jiejun Tan, Haonan Chen, Haoqi Gu, Ruofei Lai, Xinyu Zhang, Zhao Cao, and Ji-Rong Wen. 2023. Optimizing factual accuracy in text generation through dynamic knowledge selection. *arXiv preprint arXiv:2308.15711*.

Yujia Qin, Zihan Cai, Dian Jin, Lan Yan, Shihao Liang, Kunlun Zhu, Yankai Lin, Xu Han, Ning Ding, Huadong Wang, et al. 2023. Webcpm: Interactive web search for chinese long-form question answering. *arXiv preprint arXiv:2305.06849*.

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. *The Journal of Machine Learning Research (JMLR)*, 21(140).

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-context retrieval-augmented language models. *arXiv preprint arXiv:2302.00083*.

Hannah Rashkin, Vitaly Nikolaev, Matthew Lamm, Lora Aroyo, Michael Collins, Dipanjan Das, Slav Petrov, Gaurav Singh Tomar, Iulia Turc, and David Reitter. 2023. Measuring attribution in natural language generation models. *Computational Linguistics*, pages 1–64.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilic, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, Jonathan Tow, Alexander M. Rush, Stella Biderman, Albert Webson, Pawan Sasanka Ammanamanchi, Thomas Wang, Benoît Sagot, Niklas Muennighoff, Albert Villanova del Moral, Olatunji Ruwase, Rachel Bawden, Stas Bekman, Angelina McMillan-Major, Iz Beltagy, Huu Nguyen, Lucile Saulnier, Samson Tan, Pedro Ortiz Suarez, Victor Sanh, Hugo Laurençon, Yacine Jernite, Julien Launay, Margaret Mitchell, Colin Raffel, Aaron Gokaslan, Adi Simhi, Aitor Soroa, Alham Fikri Aji, Amit Alfassy, Anna Rogers, Ariel Kreisberg Nitzav, Canwen Xu, Chenghao Mou, Chris Emezue, Christopher Klamm, Colin Leong, Daniel van Strien, David Ifeoluwa Adelani, and et al. 2022. BLOOM: A 176b-parameter open-access multilingual language model. *CoRR*, abs/2211.05100.

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*.

Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. 2021. Retrieval augmentation reduces hallucination in conversation. *arXiv preprint arXiv:2104.07567*.

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, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 8273–8288. Association for Computational Linguistics.

Hao Sun, Yang Li, Liwei Deng, Bowen Li, Binyuan Hui, Binhua Li, Yunshi Lan, Yan Zhang, and Yongbin Li. 2023a. History semantic graph enhanced conversational kbqa with temporal information modeling. *arXiv preprint arXiv:2306.06872*.

Hao Sun, Xiao Liu, Yeyun Gong, Yan Zhang, Daxin Jiang, Linjun Yang, and Nan Duan. 2023b. Allies: Prompting large language model with beam search.

Ross Taylor, Marcin Kardas, Guillem Cucurull, Thomas Scialom, Anthony Hartshorn, Elvis Saravia, Andrew Poulton, Viktor Kerkez, and Robert Stojnic. 2022. Galactica: A large language model for science. *CoRR*, abs/2211.09085.

Yile Wang, Peng Li, Maosong Sun, and Yang Liu. 2023a. Self-knowledge guided retrieval augmentation for large language models. *arXiv preprint arXiv:2310.05002*.

Zhiruo Wang, Jun Araki, Zhengbao Jiang, Md Rizwan Parvez, and Graham Neubig. 2023b. Learning to filter context for retrieval-augmented generation. *arXiv preprint arXiv:2311.08377*.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. C-pack: Packaged resources to advance general chinese embedding.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. 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, Brussels, Belgium, October 31 - November 4, 2018*, pages 2369–2380. Association for Computational Linguistics.

Wenhao Yu, Dan Iter, Shuohang Wang, Yichong Xu, Mingxuan Ju, Soumya Sanyal, Chenguang Zhu, Michael Zeng, and Meng Jiang. 2022. Generate rather than retrieve: Large language models are strong context generators. *arXiv preprint arXiv:2209.10063*.

Wenhao Yu, Hongming Zhang, Xiaoman Pan, Kaixin Ma, Hongwei Wang, and Dong Yu. 2023a. Chain-of-note: Enhancing robustness in retrieval-augmented language models. *arXiv preprint arXiv:2311.09210*.

Wenhao Yu, Zhihan Zhang, Zhenwen Liang, Meng Jiang, and Ashish Sabharwal. 2023b. Improving language models via plug-and-play retrieval feedback. *arXiv preprint arXiv:2305.14002*.

Zichun Yu, Chenyan Xiong, Shi Yu, and Zhiyuan Liu. 2023c. Augmentation-adapted retriever improves generalization of language models as generic plug-in. *arXiv preprint arXiv:2305.17331*.

Jiaxin Zhang, Zhuohang Li, Kamalika Das, Bradley Malin, and Sricharan Kumar. 2023a. Sac3: Reliable hallucination detection in black-box language models via semantic-aware cross-check consistency: Reliable hallucination detection in black-box language models via semantic-aware cross-check consistency. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 15445–15458.

Yunxiang Zhang, Muhammad Khalifa, Lajanugen Logeswaran, Moontae Lee, Honglak Lee, and Lu Wang. 2023b. Merging generated and retrieved knowledge for open-domain qa. *arXiv preprint arXiv:2310.14393*.

Xinran Zhao, Hongming Zhang, Xiaoman Pan, Wenlin Yao, Dong Yu, and Jianshu Chen. 2023. Thrust: Adaptively propels large language models with external knowledge. *arXiv preprint arXiv:2307.10442*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric. P Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena.

Algorithm 1 The pipeline of VTG

Input: Question q, document pool $\mathbb{D}$, the Generators, the Verifiers, the Citation Simplifier, the Evidence Finder, the maximum trials T, the retriever R, the number of initially retrieved documents k, the number of generated queries M, the number of documents retrieved per query N
Output: Output with citations O

$t \leftarrow 0$
$O \leftarrow \{\}$
$D_S \leftarrow \{\}$
$D_L \leftarrow R(q, \mathbb{D}, k)$
while TRUE **do**
    $s \leftarrow \text{ClaimGenerator}(O, D_S \cup D_L)$
    $C \leftarrow \text{CitationGenerator}(s, D_S \cup D_L)$
    if s is <EOS> **then**
        break
    end if
    if $\text{GenerationVerifier}(s, C) \rightarrow$ TRUE **then**
        $C \leftarrow \text{CitationSimplifier}(s, C)$
        $O \leftarrow O \cup \{s, C\}$
        $D_L \leftarrow D_L \cup C$
        $t \leftarrow 0$
    else if $\text{MemoryVerifier}(s, D_S \cup D_L) \rightarrow$ TRUE **then**
        $C \leftarrow \text{CitationSimplifier}(s, D_S \cup D_L)$
        $O \leftarrow O \cup \{s, C\}$
        $D_L \leftarrow D_L \cup C$
        $t \leftarrow 0$
    else if $t > T$ **then**
        $O \leftarrow O \cup \{s, C\}$
        $t \leftarrow 0$
    else
        $D_S \leftarrow \text{EvidenceFinder}(s, M, N)$
        $t \leftarrow t + 1$
    end if
end while
return O

A Baselines

For an equitable comparison, we have selected four best-performing baseline methodologies as proposed in ALCE (Gao et al., 2023b), which include VANILLA, SUMM, SNIPPET and RERANK. Each of these methods incorporates multiple demonstrations within the initial prompt to facilitate the process of generating responses. Following ALCE (Gao et al., 2023b), we set $k = 5$ and $K = 10$ in our experiment.

VANILLA. This configuration involves providing the LLM with the top-k ranked documents. The LLM is then tasked with generating responses that appropriately include citations.

SUMM. In this approach, the LLM is required to synthesize relevant information from the top-K ranked documents. After summarizing these documents, the condensed text is integrated into the prompt. The LLM is then instructed to create texts that incorporate citations, drawing from this summarized content.

SNIPPET. In this setup, the LLM is instructed to extract relevant snippets from the top-K ranked documents. These concise documents are subsequently utilized in the prompt, with the aim for the LLM to create text that includes citations, drawing from these brief extracts.

RERANK. This methodology entails a two-stage process. Initially, the LLM generates four distinct responses based on the top-k ranked documents using high temperature. Thereafter, each response undergoes an evaluation for citation recall. The response with the highest citation recall score is then chosen as the final output.

POSTCITE: The LLM first generates answers then cites the most relevant passage from the top-100 retrieved documents for each statement.

REFINECITE: The LLM first produces answers with citations, then uses NLI to remove unnecessary citations, refining the citation set.

VERICITE: The LLM first generates sentences with citations and then uses an NLI method to ensure the citation set conclusively supports the sentence. If not, it finds and cites the best matching passage from the top-100 retrieved documents using GTR.

VERIREFINE: Integrates RefineCite's refinement and VeriCite's verification processes, ensuring citations are both necessary and fully supportive, optimizing citation accuracy and relevance

The complete experimental result is shown in Table 8 and Table 9.

B Datasets and Settings

Datasets and experimental settings are summarized in Table 5.

Citation Recall. Citation recall for each claim in the model's response is computed individually as either 0 or 1 and then averaged across all claims in the response. A claim's citation recall is 1 if at least one citation exists and the concatenated citations entail the claim according to an NLI model, which outputs 1 for entailment.

Citation Precision. Citation precision for each citation in the model's response is computed individually as either 0 or 1 and then averaged across all citations in the response. The precision score for a citation is 1 if the associated claim has a citation recall of 1 and the citation is not irrelevant;

Settings	2WikiMultihopQA (Ho et al., 2020)	ASQA (Stelmakh et al., 2022)	ELI5 (Fan et al., 2019b)	NQ (Kwiatkowski et al., 2019)	WebQ (Berant et al., 2013)
Dataset statistics					
Task	Multihop QA	Long-form QA	Long-form QA	Open-domain QA	Open-domain QA
#Examples (Vicuna)	500	500	500	500	500
#Examples (Davinci)	200	200	200	200	200
Evaluation settings					
Correctness Metrics	EM, Token-level F1, Pre, and Rec	EM, Disambig-F1, ROUGE, DR	Claim	EM, F1	EM, F1
Citation Metrics	Prec, Rec, F1				
Retrieval settings					
Corpus	Wikipedia	Wikipedia	Wikipedia	Wikipedia	Wikipedia
Retriever	DPR	DPR	DPR	DPR	DPR

Table 5: Statistics and experimental settings of different tasks/datasets.

Parameter	MultihopQA	ASQA	ELI5	NQ	WebQ
Maximum trials T	5	2	3	3	3
Generated queries number M	2	4	2	2	4
Initially retrieved documents number k	5	5	5	5	5
Retrieved documents number per query N	4	2	2	3	2

Table 6: Hyper-parameters for VTG on `Text-Davinci-003`, where MultihopQA refers to 2WikiMultihopQA.

otherwise, it's 0. A citation is deemed irrelevant if: (a) the citation alone cannot substantiate the claim, and (b) omitting the citation doesn't impact the remaining citations' ability to support the claim.

LLM Evaluation To ensure a fair comparison of citation quality, we instruct Qwen-MAX to evaluate model generations. Specifically, we assess the quality of the citations in two ways: (1) Citation Recall: The large language model (LLM) is given a sentence and all the passages that the sentence cited, and is asked to judge whether the passages fully support the sentence; (2) Citation Precision: Given a sentence and one of its citations, the LLM is asked to judge whether the citation "fully supports" or "does not support" the sentence. Each citation receives a precision score of 1 if the output sentence has a citation recall of 1 and this citation is "fully support."

Focus Shifting Phenomenon To analyze the focus shifting phenomenon within the datasets, we employ BGE embedding[6](Xiao et al., 2023) to represent each sentence from the LLM's outputs for all questions. Subsequently, we calculate the cosine similarity between these embedded sentences to construct a similarity matrix for each question. An average of all these similarity matrices is computed, and the results are depicted in Figure 5.

The color intensity on the diagonal of the matrix is the strongest, which signifies a high degree of similarity between each sentence and itself. Moreover, there is a noticeable gradation in color intensity as the distance between the current and the target sentences increases, indicating a decrease in similarity. This pattern illustrates that the content focus of the sentences tends to diverge significantly as the LLM continues the generation, which is also called the focus shifting phenomenon(Lan and Jiang, 2021; Sun et al., 2023a).

C Hyper-parameters

The detailed hyper-parameters used in VTG for `Text-Davinci-003` and `Vicuna-13B-v1.5-16k` are shown in Table 6 and Table 7, respectively.

D Algorithm

The algorithm procedural of VTG is shown in Algorithm 1

E Prompts

The prompts used in our experiments are listed as follows. It's worth noting that the prompts for VANILLA and RERANK are identical, so we only present the one for VANILLA.

[6] https://huggingface.co/BAAI/bge-base-en-v1.5

Parameter	MultihopQA	ASQA	ELI5	NQ	WebQ
Maximum trials T	5	3	5	3	3
Generated queries number M	4	2	2	3	3
Initially retrieved documents number k	5	5	5	5	5
Retrieved documents number per query N	2	4	4	3	3

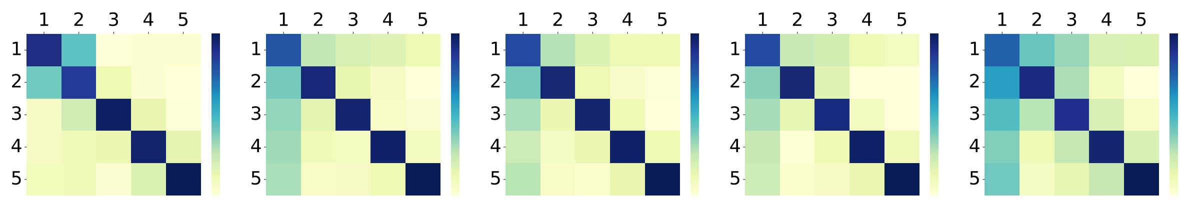
Table 7: Hyper-parameters for VTG on `Vicuna-13B-v1.5-16k`, where MultihopQA refers to 2WikiMultihopQA.

Datasets	Wikihop								WebQ								NQ							
	Correct		ALCE.Citation			LLM.Citation			Correct	ALCE.Citation			LLM.Citation			Correct	ALCE.Citation			LLM.Citation			
Metrics	EM	F1	Rec	Prec	F1	Rec	Prec	F1	EM	Rec	Prec	F1	Rec	Prec	F1	EM	Rec	Prec	F1	Rec	Prec	F1	
									Vicuna-13B														
VANILLA	23.40	21.98	29.55	22.25	25.39	41.59	35.04	38.03	55.80	67.66	60.66	63.97	67.50	67.83	67.67	54.80	71.39	61.71	66.20	77.46	63.94	70.05	
SUMM	23.20	20.00	30.89	28.43	29.61	37.66	39.12	38.37	58.00	70.51	62.07	66.02	68.23	66.57	67.39	57.00	51.55	52.21	51.88	56.76	62.86	59.66	
SNIPPET	21.80	20.05	25.18	21.95	23.45	33.67	29.75	31.59	58.40	53.44	49.15	51.21	68.46	69.57	69.01	57.20	43.56	41.43	42.47	57.56	59.57	58.55	
RERANK	22.60	21.13	47.03	47.53	47.28	54.33	53.73	54.03	56.40	89.93	76.33	82.57	88.20	67.77	76.64	56.20	83.56	73.57	78.25	81.66	73.49	77.36	
POSTCITE	22.20	15.01	8.34	6.90	7.55	22.59	15.76	18.57	58.60	46.26	32.73	38.34	65.66	61.11	63.30	46.60	35.21	24.40	28.83	52.88	50.98	51.91	
REFINECITE	24.20	22.15	48.00	39.50	43.34	52.61	47.12	49.72	55.40	78.04	72.55	75.19	84.44	79.72	82.01	55.20	65.05	61.80	63.39	75.47	77.00	76.23	
VERICITE	22.89	21.29	14.69	8.21	10.53	41.36	24.77	30.98	57.20	80.97	54.41	65.08	78.97	69.07	73.69	54.80	75.81	54.17	63.19	83.81	65.83	73.74	
VERIREFINE	24.61	23.06	15.90	11.30	13.21	47.57	32.97	38.94	54.40	82.33	64.67	72.44	81.33	75.00	78.04	55.60	71.34	58.35	64.19	81.34	67.52	73.79	
VTG	**25.60**	**23.27**	**55.36**	**49.59**	**52.32**	**62.76**	**54.69**	**58.45**	**60.00**	**92.16**	**86.51**	**89.25**	**89.43**	**81.38**	**85.21**	**58.00**	**88.69**	**82.02**	**85.22**	**86.35**	**78.06**	**82.00**	
									Text-Davinci-003														
VANILLA	33.00	33.01	40.46	28.30	33.30	59.07	43.00	49.77	67.50	63.78	58.97	61.28	71.18	66.52	68.77	62.50	60.48	55.56	57.92	66.45	61.59	63.93	
SUMM	30.00	30.63	9.39	12.19	10.61	23.19	24.64	23.89	67.50	60.06	47.62	53.12	68.33	56.65	61.95	62.50	44.23	38.45	41.14	55.36	49.20	52.10	
SNIPPET	32.00	30.13	13.86	18.49	15.84	37.36	38.99	38.16	67.00	65.41	52.32	58.14	71.81	68.07	69.89	62.00	54.72	46.99	50.56	73.05	69.55	71.25	
RERANK	32.67	33.09	56.13	45.22	50.09	63.43	46.32	53.54	67.00	73.72	64.90	69.03	78.12	70.00	73.84	61.50	71.30	63.44	67.14	79.03	66.37	72.15	
VTG	**41.50**	**40.19**	**63.89**	**57.65**	**60.61**	**70.47**	**59.13**	**64.30**	**68.00**	**93.00**	**88.72**	**90.81**	**90.70**	**87.52**	**89.08**	**63.00**	**91.85**	**86.59**	**89.14**	**84.92**	**73.20**	**78.63**	

Table 8: Comparisons between VTG and baselines on Multi-hop QA task and Open-domain QA task.

Datasets	ASQA										ELI5							Overall	
	Correct				ALCE.Citation			LLM.Citation			Correct	ALCE.Citation			LLM.Citation			Correct	Citation
Metrics	EM	D-F1	R-L	DR	Rec	Prec	F1	Rec	Prec	F1	Claim	Rec	Prec	F1	Rec	Prec	F1	EM	F1
									Vicuna-13B										
VANILLA	32.00	27.52	33.53	30.53	72.78	62.09	67.01	73.28	66.59	69.78	12.20	59.79	48.26	53.41	81.46	76.89	79.11	35.64	60.06
SUMM	41.71	28.95	37.18	33.07	62.15	59.60	60.85	68.95	70.19	69.56	14.20	60.13	52.42	56.01	77.87	72.42	75.05	38.82	57.44
SNIPPET	39.22	27.01	35.65	31.33	46.23	47.04	46.63	56.55	63.03	59.61	14.33	31.47	32.72	32.08	46.69	49.32	47.97	38.19	46.26
RERANK	37.14	28.21	32.18	30.20	88.29	75.74	81.53	88.29	75.74	81.53	11.67	73.80	61.12	66.86	84.57	77.09	80.65	36.80	72.67
POSTCITE	25.55	22.24	35.36	28.80	38.54	25.69	30.83	47.14	38.14	42.16	14.47	26.90	17.02	20.85	62.15	52.20	56.74	33.48	35.91
REFINECITE	36.30	28.24	35.35	31.79	77.80	60.50	68.07	69.80	61.17	65.20	11.60	26.90	59.05	36.96	32.15	67.46	43.54	36.54	60.36
VERICITE	34.36	28.86	36.16	32.51	76.15	72.67	74.37	77.32	75.83	76.57	13.73	64.93	40.14	49.61	77.93	62.48	69.35	36.60	58.71
VERIREFINE	34.36	28.86	35.40	32.13	80.80	59.28	68.39	76.80	64.95	70.38	11.27	61.57	37.59	46.68	64.57	51.26	57.15	36.05	58.32
VTG	**41.92**	**30.53**	**37.87**	**34.20**	**89.15**	**82.57**	**85.73**	**89.15**	**82.57**	**85.73**	**14.73**	**81.50**	**72.16**	**76.55**	**87.60**	**84.46**	**86.00**	**40.05**	**78.65**
									Text-Davinci-003										
VANILLA	40.25	31.47	35.81	33.64	58.13	55.17	56.61	58.13	55.17	56.61	13.43	58.66	47.40	52.43	58.66	47.40	52.43	43.34	55.31
SUMM	41.33	28.91	37.21	33.06	48.31	40.68	44.17	50.48	44.44	47.27	11.50	39.43	31.81	35.21	52.27	48.47	50.30	42.57	41.98
SNIPPET	39.60	30.11	38.35	34.23	53.14	43.19	47.65	59.31	52.05	55.44	13.67	45.29	37.23	40.87	62.39	55.19	58.57	42.85	50.64
RERANK	39.55	29.94	39.38	34.66	75.83	69.81	72.70	76.41	70.01	73.07	14.76	76.21	61.67	68.17	86.98	77.64	82.04	43.10	68.18
VTG	**41.53**	**31.64**	**39.45**	**35.55**	**86.70**	**79.95**	**83.19**	**89.10**	**79.84**	**84.22**	**16.67**	**82.63**	**71.56**	**76.70**	**87.94**	**81.79**	**84.75**	**46.14**	**80.14**

Table 9: Comparisons between VTG and baselines on Long-form QA task and overall performance.



(a) Wikihop (b) WebQ (c) NQ (d) ASQA (e) ELI5

Figure 5: The performance change over different hyper-parameters on various datasets.

Prompt for Sentence Generator

Instruction: Write an accurate, engaging, and concise answer for the given question using only the provided search results (some of which might be irrelevant).
Question: {Question}
Document: {Document}
Answer:

Prompt for Citation Generator

Instructions: You will be provided with a sentence and several related documents. Your task is to directly append citation annotations to the sentence using these documents without changing the sentence. When citing documents, use [1][2][3]. Cite at least one document and at most three documents. If multiple documents support the sentence, only cite a minimum sufficient subset of the documents.
Document: {Document}
Sentence: {Sentence}
Sentence with citation:

Prompt for Query Generation

Given the original question: {Question}.
The context is as follows: {Context}.
The claim is: {Claim}.
Please generate up to {qg_num} questions that can help verify the claim with the following constraints:
1. You should output no more than {qg_num} questions.
2. The generated questions should be diverse and focus on different aspects of the given claim.
Generated questions:

Prompt for VANILLA

Instruction: Write a high-quality answer for the given question using only the provided search results and cite them properly using [1][2][3].
Question: {Question}
Document: {Document}
Answer:

Prompt for SUMM

Step 1: First Summarize the documents
Summarize the following document within 50 words with the question of interest {Question}
Return "irrelevant" if the document is “irrelevant" to the question. Try to keep all the important dates, numbers, and names.
Title: {Title}
Text: {Text}
Summary:

Step 2: Generate the response based on the summary
Instruction: Write a high-quality answer for the given question using only the provided search results and cite them properly using [1][2][3].
Question: {Question}
Document: {Document}
Answer:

Prompt for SNIPPET

Step 1: First extract relevant snippet from the documents
Given the following passage and the question {Question}, extract a useful span from the passage that can answer the question. Resolve all the coreference issues to make the extracted span understandable and standalone. If the passage is not helpful for answering the question, return “irrelevant". If there are multiple spans, merge them and only output one paragraph.
Title: {Title}
Text: {Text}
Extracted span:

Step 2: Generate the response based on the snippet
Instruction: Write a high-quality answer for the given question using only the provided search results and cite them properly using [1][2][3].
Question: {Question}
Document: {Document}
Answer:

Prompt for Citation Evaluation

Role: Data Annotator
Instructions:
You are provided with the following materials:
- **Passage**: passage
- **Sentence**: sentence
Task: Assess whether the passage fully supports the sentence.
Choices:
1. **Fully Supports**: Select this option if the passage completely and clearly supports every aspect of the sentence.
2. **Does Not Fully Support**: Select this option if any discrepancies, omissions, or inaccuracies in the passage prevent it from fully supporting the sentence.
Output:
- If the passage fully supports the sentence, output "Yes."
- If it does not, output "No."
Note: Please refrain from adding any content not requested in the instructions.