

VeriCite: Towards Reliable Citations in Retrieval-Augmented Generation via Rigorous Verification

Haosheng Qian, Yixing Fan*

State Key Laboratory of AI Safety,
ICT, CAS

University of Chinese Academy of
Sciences, CAS
Beijing, China

qianhaosheng22@mails.ucas.ac.cn
fanyixing@ict.ac.cn

Jiafeng Guo

State Key Laboratory of AI Safety,
ICT, CAS

University of Chinese Academy of
Sciences, CAS
Beijing, China

guojiafeng@ict.ac.cn

Ruqing Zhang

State Key Laboratory of AI Safety,
ICT, CAS

University of Chinese Academy of
Sciences, CAS
Beijing, China

zhangruqing@ict.ac.cn

Qi Chen

Meituan Inc.

Beijing, China

cqict90@gmail.com

Dawei Yin

Baidu Inc.

Beijing, China

yindawei@acm.org

Xueqi Cheng

State Key Laboratory of AI Safety,
ICT, CAS

University of Chinese Academy of
Sciences, CAS
Beijing, China

cxq@ict.ac.cn

Abstract

Retrieval-Augmented Generation (RAG) has emerged as a crucial approach for enhancing the responses of large language models (LLMs) with external knowledge sources. Despite the impressive performance in complex question-answering tasks, RAG still struggles with hallucinations. Attributing RAG-generated content through in-line citations has demonstrated potential in reducing hallucinations and facilitating human verification. Existing citation generation methods primarily rely on either fine-tuning the generator or employing post-processing approaches for citation matching. However, the former approach demands substantial annotated data and computational resources, while the latter often encounters difficulties in managing multiple citations and frequently produces suboptimal results. In this paper, we introduce a novel framework, called **VeriCite**, designed to rigorously validate supporting evidence and enhance answer attribution. Specifically, **VeriCite** breaks down into a three-stage generation: 1) The *initial answer generation* first generates a response based on all available contexts and has its claims verified through the NLI model; 2) the *supporting evidence selection* assesses the utility of each document and extracts useful supporting evidences; 3) the *final answer refinement* integrates the initial response and collected evidences to produce the final, refined answer. We conduct experiments across five open-source LLMs and four datasets, demonstrating that VeriCite can significantly improve citation quality while maintaining the correctness of the answers.¹

*Corresponding author.

¹Our code is publicly available at <https://github.com/QianHaosheng/VeriCite>

CCS Concepts

• **Information systems** → **Information systems applications**.

Keywords

Large Language Model, Retrieval-Augmented Generation, Response Attribution

ACM Reference Format:

Haosheng Qian, Yixing Fan, Jiafeng Guo, Ruqing Zhang, Qi Chen, Dawei Yin, and Xueqi Cheng. 2025. VeriCite: Towards Reliable Citations in Retrieval-Augmented Generation via Rigorous Verification. In *Proceedings of the 2025 Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region (SIGIR-AP 2025)*, December 7–10, 2025, Xi'an, China. ACM, New York, NY, USA, 8 pages. <https://doi.org/10.1145/3767695.3769505>

1 Introduction

Retrieval-Augmented Generation (RAG) [9, 16, 30] plays a crucial role in enabling large language models (LLMs) [1, 19] to tackle challenges such as real-time news queries and domain-specific issues, thereby expanding the capabilities and application scope of LLMs. However, as retrieval technology is not always flawless, it simultaneously introduces new challenges for LLMs. For example, if irrelevant information is retrieved and used as a reference, the LLM may incorporate this noise and generate incorrect answers, exacerbating the hallucination issue [6, 15, 35].

Therefore, enabling LLMs to generate attributable responses is vital for ensuring trustworthiness and mitigating misinformation. An effective strategy to enhance the reliability of LLM responses is through citation mechanisms, whereby each statement is explicitly anchored to relevant source materials [5, 8, 17]. This approach not only establishes traceability by allowing users to independently verify the accuracy of responses, but also facilitates error diagnosis and promotes transparency in human-AI collaboration.



This work is licensed under a Creative Commons Attribution 4.0 International License. *SIGIR-AP 2025, Xi'an, China*

© 2025 Copyright held by the owner/author(s).

ACM ISBN 979-8-4007-2218-9/2025/12

<https://doi.org/10.1145/3767695.3769505>

Current approaches for generating answers with citations can be broadly classified into two paradigms [14]. The first category, classified as “intrinsic attribution”, operates synchronously with text generation. These approaches typically treat citations as regular tokens and enable LLM to directly generate citations within answers through fine-tuning or in-context learning [8, 20, 23]. Nevertheless, intrinsic integration approaches face several practical constraints: (1) Fine-tuning demands extensive domain-specific annotation and significant computational resources; (2) In-context learning is highly sensitive to the input examples, leading to poor generalization performance.

The other category can be classified as “extrinsic attribution”, which initially generates a draft answer and subsequently employs post-processing approaches to match retrieved passages with statements in the answer. Common matching methods include utilizing sentence similarity metrics such as BLEU [24] and ROUGE [18], or employing Natural Language Inference (NLI) classifiers to evaluate entailment relationships [7]. Classic similarity metrics are computationally efficient, but their effectiveness is constrained by the challenge of determining thresholds. Conversely, although NLI models deliver higher accuracy, yet fundamentally struggle to handle cases where a single statement requires multiple citations [8].

To address the aforementioned issues, we propose a novel framework named **VeriCite**, which strengthens the reliability of citations through rigorous verification. In contrast to previous studies which primarily focused on the answer generation process or post-processing stages, VeriCite concentrates on the phase after retrieved passages are obtained but before final answer generation commences. VeriCite consists of three stages: initial answer generation, supporting evidence selection, and final answer refinement (as illustrated in Figure 1). The initial answer generation stage generates a response based on all retrieval passages and uses an NLI model to verify the citations in the statement, ensuring the reliability of the answer. The subsequent supporting evidence selection stage thoroughly extracts potentially useful evidence from each passage. This evidence must also undergo verification through the NLI model, and the verified evidence is then marked with citations. The final answer refinement stage integrates the initial answer and the extracted evidence, with the LLM responsible for reorganizing the order of the statements to improve fluency, removing redundant content, and merging citations.

VeriCite aims to pre-screen the content within retrieved passages that is genuinely valuable for answer generation, pre-attributing citations to these high-quality segments to ensure source traceability. This preprocessing approach helps eliminate noise from the input, significantly alleviating the cognitive load on LLMs when extracting key information from long contexts. Furthermore, the strategy of pre-attributing citations reduces the model’s attribution difficulty, enabling the generator to more seamlessly and accurately reuse existing citations within the answer. Extensive experiments conducted across multiple datasets and LLMs demonstrate that while achieving answer accuracy on par with baselines, VeriCite yields a significant improvement in citation generation quality.

2 Related Work

In retrieval-augmented question answering, methods for generating attributed answers typically fall into two primary categories.

The first category employs “intrinsic attribution”, leveraging generative models’ inherent attribution capabilities. This approach typically utilizes supervised fine-tuning or in-context learning to enable models to produce answers with integrated citations. Among seminal implementations, WebGPT [23] enhances open-domain question answering (QA) accuracy by simulating human web browsing behavior. Built upon GPT-3 [1], this system extracts relevant webpage passages as supporting evidence and inserts citations as commandS within answers. The authors trained reward models on extensive human preference annotations, optimizing answer quality through Proximal Policy Optimization [29]. Subsequent innovations include WebGLM [20], which integrates an LLM-augmented retriever, bootstrapped generator, and a human preference-aware scorer. Its automated annotation pipeline enabled large-scale training data generation, with supervised fine-tuning on citation-annotated QA data yielding robust attribution capabilities. APO [17] advances training methodology by formulating attribution as preference learning and introducing a progressive optimization framework with sentence-level rewards that enhances alignment efficiency. Distinctively, LongCite [42] tackles fine-grained citation in long-context QA through its Coarse to Fine (CoF) data construction scheme, enabling precise sentence-level attribution with superior traceability relative to passage-level alternatives. Unlike approaches requiring fine-tuning, alternative approaches employ prompting to instruct models to incorporate citations during answer generation. ALCE [8] systematically evaluated multiple few-shot citation generation strategies, including Vanilla, Summary, and Snippet. Alternative research efforts have designed more sophisticated reasoning pipelines dedicated to enabling models to perform proactive verification and citation refinement during generation. For instance, VTG [32] introduces a document storage mechanism with long short-term memory, implements an active retrieval component that generates diversified queries, and incorporates a hierarchical verification module featuring an evidence finder to validate relationships between generated answers and their citations.

Contrastingly, “extrinsic attribution” methods incorporate citations during post-processing. These methods first generate an initial answer (with or without citations) using a generative model, then establish correspondences between the generated text and retrieved passages through text matching techniques, and finally insert appropriate citations [11]. This strategy enables attribution even for models lacking inherent citation capabilities. For instance, WebGLM’s automated citation annotation pipeline utilized the ROUGE-1 [18] similarity metric to evaluate citation correctness, filtering higher-quality training data. Beyond text similarity metrics, alternative approaches leverage NLI models to determine entailment relationships between answer sentences and retrieved passages, assigning citations based on classification results. ALCE implemented this NLI approach as a representative post-processing baseline method.

Effective evaluation methodologies are indispensable for advancing citation generation research, with established approaches encompassing both human assessment and automated metrics [25, 38].

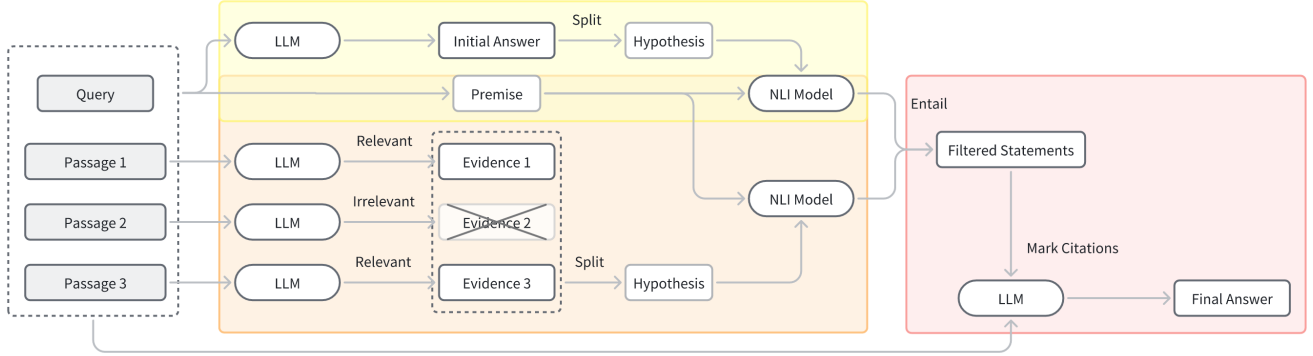


Figure 1: Overview of VeriCite framework.

The pioneering human evaluation framework, Attributable to Identified Sources (AIS) [27], measures textual faithfulness to source materials through a structured protocol: annotators first examine model-generated text to determine whether each statement requires external source substantiation, then verify (1) the presence of explicit source attribution, and (2) content consistency between generated claims and corresponding source materials. While human evaluation offers superior accuracy, its significant drawbacks include high labor costs and low efficiency. To address these challenges, researchers proposed AutoAIS [7] based on AIS, which leverages NLI models to approximate human judgment. This automated approach refines evaluation granularity to the sentence level by examining entailment relationships between responses and source materials. Building upon this foundation, ALCE redefines citation recall and citation precision metrics while establishing the first benchmark for LLM attribution evaluation. This benchmark incorporates multi-dimensional evaluation of fluency, correctness, and citation quality. Further advancing the field, CAQA [13] introduces a comprehensive four-category framework (Supported, Insufficient, Contradictory, Irrelevant) for fine-grained attribution evaluation, enabling more precise quantification of attribution performance.

3 VeriCite framework

3.1 Task Formulation

Following previous work [8, 20], the formal description of this task is as follows: Given a query q , top- k passages $P = \{p_1, p_2, \dots, p_k\}$ are retrieved, and the LLM needs to generate an answer A . The answer A consists of several statements $A = \{s_1, s_2, \dots\}$. Each statement s_i may cite a set of passages $C_i = \{c_{i1}, c_{i2}, \dots\}$, where $c_{ij} \in [1, k] \cap \mathbb{Z}$.

3.2 Initial Answer Generation

The initial answer generation phase follows standard RAG methodology [16], where the query q and all top- k retrieved passages $\{p_1, p_2, \dots, p_k\}$ are concatenated into a single input sequence for the LLM, producing an initial answer $init_ans$.

$$init_ans = Answer(q, p_1, p_2, \dots, p_k) \quad (1)$$

At this stage, we employ a few-shot instruction template to provide in-context learning examples, guiding the LLM to learn the citation patterns demonstrated in the exemplars. This approach explicitly requires the model to incorporate citations within each answer statement. The instruction template at this stage is shown in Appendix A.1. This phase aims to produce a foundational answer for subsequent refinement, requiring only citation incorporation without additional model constraints. While contemporary LLMs excel at answering simple commonsense queries, they inevitably exhibit hallucination tendencies when confronted with complex problems.

To enhance answer reliability, we implement a rigorous verification and filtering mechanism. During initial answer validation, unsupported content must be systematically eliminated, retaining exclusively evidence-substantiated answer statements. To facilitate granular reliability verification, the initial answer $init_ans$ is decomposed into a set of statements $\{s_1, s_2, \dots\}$, where each statement s_i is associated with a potentially empty set of citations $\{c_{i1}, c_{i2}, \dots\}$. The verification process employs a NLI model ϕ trained for Recognizing Textual Entailment (RTE) tasks [2, 41], which predicts whether a hypothesis is entailed by, contradicts, or is neutral to given premises. Specifically, for each statement s_i , we validate whether the corresponding retrieval passages $\{p_{c_{ij}}\}$ (premises) entail the statement s_i (hypothesis).

$$sup_i = \phi(\text{concat}(p_{c_{ij}}), s_i) \quad (2)$$

The verification outcome sup_i is binary-valued: when the model determines that the answer statement s_i is entailed by (a combination of) the retrieved passages, sup_i is assigned *True*; otherwise, it returns *False*, indicating an unsupported statement likely containing hallucinated content that consequently fails verification and must be discarded.

3.3 Supporting Evidence Selection

Irrelevant information can interfere with the LLM's response generation, potentially causing critical relevant details to be overlooked. While conventional RAG approaches generate answers based on coarsely aggregated retrieval results, our methodology additionally incorporates fine-grained evidence extraction. This necessity

-	Settings	ASQA	ELI5	HotpotQA	MuSiQue
	Task	Long-form QA	Long-form QA	Multihop QA	Multihop QA
Dataset statistics	Question Type	Factoid	How/Why/What	Factoid	Factoid
	# Examples	948	1000	500	500
Evaluation metrics	Correctness	EM Recall	Claim Recall	EM Recall	EM Recall
	Citation Quality	Citation Recall, Citation Precision, Citation F1			

Table 1: Statistics of different datasets.

arises from the fundamental misalignment between retriever and generator objectives in standard RAG pipelines [21]. These distinct models often exhibit mutually incompatible relevance judgments. Retrievers may introduce either irrelevant information or seemingly relevant but non-actionable content, both of which can cause generators to overlook genuinely critical information while producing noise-degraded outputs.

Inspired by recent studies [26, 28, 33], our evidence selection phase leverages the LLM’s robust natural language understanding capabilities to collaborate with the retriever in context extraction. This dual engagement strategy comprehensively excavates potentially valuable content that might otherwise be overlooked within each passage. Specifically, the LLM first independently evaluates each retrieved passage’s utility for answering the query using the instruction template depicted in Appendix A.2.

$$rel_i = \text{Check}(q, p_i) \quad (3)$$

Following the LLM’s secondary verification of passage utility, retained passages ($rel_i = \text{True}$) proceed to evidence selection. The generator then independently produces answers for each qualifying passage p_i using the instruction template shown in Appendix A.3.

$$evidence_i = \begin{cases} \text{Answer}(q, p_i) & , rel_i = \text{True} \\ \text{None} & , rel_i = \text{False} \end{cases} \quad (4)$$

Like other texts generated by LLMs, $evidence_i$ remains prone to hallucination issues, necessitating further verification of its entailment relationship with the corresponding original passage p_i . Similar to the previous phase, the verification process decomposes $evidence_i$ into statements $\{s_{i1}, s_{i2}, \dots\}$. We then employ the NLI model ϕ to verify whether the original retrieval passage entails these statements.

$$sup_{ij} = \phi(p_i, s_{ij}) \quad (5)$$

Statements s_{ij} verified by the NLI model as entailed by passage p_i are retained for subsequent summarization and automatically annotated with the corresponding citation marker “[i]”. This design fundamentally decouples attribution from generation during the summarization phase. Final answer citations directly reuse these pre-process markers rather than relying on the generator’s attribution capabilities, thereby significantly reducing demands on the LLM’s citation capacity.

3.4 Final Answer Refinement

Following rigorous collection and verification procedures in the preceding stages, we obtain a curated set of semantically validated statements $\{s_1, s_2, \dots\}$ accompanied by their corresponding citation

sets $\{c_1 = \{c_{11}, c_{12}, \dots\}, c_2 = \{c_{21}, c_{22}, \dots\}, \dots\}$. While these components exhibit high reliability due to rigorous verification, their inherent fragmentation and potential redundancy render them unsuitable for direct concatenation into a coherent final response. The refinement phase fundamentally redefines the large language model’s role rather than directly addressing the query or making attribution decisions, the model now functions as a synthesis engine. This engine processes the verified statements and citations as foundational input material, executing three critical transformations: restructuring logical flow and sentence sequencing to enhance coherence, eliminating redundant content to improve conciseness, and strategically consolidating citations to optimize referential clarity.

$$final_ans = \text{Refine}(q, P, s_1, c_1, s_2, c_2, \dots) \quad (6)$$

To mitigate potential referential ambiguity and ensure contextual fidelity, the original retrieved passages are incorporated into the model’s input stream. This architectural choice provides essential grounding context, enabling more accurate interpretation of statement semantics and preventing summarization errors arising from ambiguous references. Furthermore, explicit instructional constraints mandate that the model preserve the original semantic content of input statements without modification while simultaneously ensuring the final output maintains both informational completeness and fluent logical progression. The model must achieve this dual objective through careful rhetorical reorganization rather than content alteration. The instruction template at this stage is shown in Appendix A.4.

4 Experiments

4.1 Datasets and Models

To comprehensively evaluate our method’s effectiveness across diverse question types, we conduct experiments on four benchmark datasets. The long-form QA datasets include ASQA [31], an ambiguity-aware factual dataset distinguished from conventional benchmarks by its exclusive focus on ambiguous questions sourced from AmbigQA [22]. Each query admits multiple valid interpretations, necessitating models to recognize inherent ambiguities and synthesize comprehensive responses using evidentiary support. Complementing this, ELI5 [4] comprises predominantly non-factual questions originating from Reddit’s “Explain Like I’m Five”² forum. Characterized by complex *how*, *why*, and *what* queries, this dataset presents significant challenges in generating logically coherent, information-rich long-form explanations. For multi-hop reasoning evaluation, we employ HotpotQA [40] and MuSiQue [36]. HotpotQA features curated factual questions requiring cross-document

²<https://www.reddit.com/r/explainlikeimfive>

evidence integration through manually designed multi-step reasoning. Conversely, MuSiQue contains synthetically generated factual questions formed by composing single-hop queries, typically demanding 2-4 inference steps. This automated composition process yields linguistically structured questions that present heightened analytical difficulty relative to conventional benchmarks.

The ASQA and ELI5 datasets are subsets released by ALCE [8], while HotpotQA and MusiQue are subsets released by IRCOT [37]. Each dataset is evaluated in terms of answer correctness and citation quality. Among these, we use the EM (Exact Match) Recall metric to evaluate the answer correctness for the ASQA, HotpotQA, and MusiQue datasets, use Claim Recall to evaluate the answer correctness for the ELI5 dataset, and use Citation F1 to evaluate the citation quality for all datasets. Dataset details are summarized in Table 1.

Experiments were conducted on five open-source LLMs: Llama3-8B-Instruct [3], Gemma-2-9B-it [34], GLM-4-9B-Chat [10], Qwen2.5-7B-Instruct [39], and Qwen2.5-14B-Instruct.

4.2 Baselines

For baseline comparisons, we selected four established approaches:

- **Vanilla** [8]: The query and top- k retrieved passages are concatenated to form the model input. Task-specific instructions coupled with in-context learning mechanisms guide the generation of answers with integrated citations. This approach represents the foundational methodology for attribution generation, processing retrieved passages without additional refinement.
- **Summary** [8]: Retrieved passages undergo summarization-based compression prior to model input. These summarized compressions are concatenated with the original query and processed through identical task-specific instructions and in-context learning mechanisms to guide the generation of answers with integrated citations. This approach intentionally mitigates textual redundancy in model inputs, enhancing focus on salient information.
- **Snippet** [8]: Contrasting with the Summary approach, this methodology employs extractive summarization for model input. This methodology preserves exact expressions from retrieved passages, thereby circumventing potential semantic distortion inherent in abstractive summarization.
- **APO** [17]: Automatic Preference Optimization framework enhances model performance through a dual-phase approach: supervised fine-tuning followed by preference optimization. During the preference optimization phase, a novel loss function is implemented to enable fine-grained sentence-level rewards, facilitating more efficient model parameter updates.

4.3 Implementation Settings

In the experiment, the top-5 retrieved passages are provided for each query, and each method is given two few-shot examples for in-context learning. In the VeriCite method, we use TRUE [12] as the NLI model for citation verification. To ensure the reproducibility of the experiment, all LLMs generate responses using greedy decoding.

4.4 Main Results

Our experimental results, as shown in Table 2.

On the ASQA dataset, VeriCite exhibits a clear advantage in answer correctness across all five models, outperforming all baseline methods. Notably, the GLM-4 model delivers the most substantial improvement with a 4.54% increase in correctness over the best performing Vanilla baseline. Regarding citation quality, Llama3 and Qwen2.5 models achieve significant enhancements in citation F1, surpassing the strongest baseline. In contrast, Gemma-2 and GLM-4 perform marginally below their respective optimal baselines in this metric.

For the ELI5 dataset, VeriCite underperforms relative to the more robust baselines in answer correctness across all five models, indicating potential limitations in its answer generation mechanism for non-factoid questions. It is noteworthy that the extensively fine-tuned APO baseline demonstrates strong correctness here, exhibiting only minor degradation with the GLM-4 model. Conversely, VeriCite achieves substantial gains in citation quality, with all five models exceeding the best baseline by an average margin of 11.41% in Citation F1 score, thereby validating its efficacy for citation optimization.

On multi-hop QA datasets, VeriCite shows a pronounced improvement in answer correctness exclusively with the Qwen2.5 model, which surpasses all other baselines. However, its performance with the remaining three models falls slightly below their respective best baselines. This observation suggests that VeriCite’s supporting evidence selection stage may be suboptimal for multi-hop scenarios requiring cross-passage information integration, highlighting a potential area for architectural refinement. Despite this, all models exhibit exceptional citation quality, significantly outperforming the strongest baselines in Citation F1 scores.

Overall, the results indicate that VeriCite matches or exceeds the best baselines in answer correctness, with particularly notable gains observed for the Qwen2.5 and GLM-4 models. Simultaneously, citation quality was significantly enhanced across all five models compared to the best baseline performances. Furthermore, both parameter scales of the Qwen2.5 model exhibited similar improvements, suggesting that the proposed method retains its potential for application to larger-scale models.

5 Analysis

5.1 Ablation Study

This section presents an ablation study conducted on the VeriCite framework to evaluation the contribution of its core components. Experiments were performed using the Llama3-8B-Instruct model on the ASQA dataset.

Three specific ablation variants were investigated. The first variant omits the initial answer generation stage; consequently, the final answer is generated exclusively utilizing statements derived from the supporting evidence selection stage. The second variant removes the supporting evidence selection stage, with the final answer organized solely based on statements obtained from the initial answer generation stage. The third variant eliminates the NLI based verification module employed in both the initial answer generation and supporting evidence selection stages. Under this condition, all generated statements are assumed to be supported

Model	Method	ASQA		ELI5		HotpotQA		MuSiQue		Overall	
		EM	Citation F1	Claim	Citation F1	EM	Citation F1	EM	Citation F1	Correct	Citation F1
Llama3-8B	Vanilla	38.41	69.48	12.80	38.33	43.60	35.76	12.20	17.33	26.16	44.35
	Summary	37.81	65.21	10.60	42.32	36.80	24.46	3.20	4.72	22.54	40.27
	Snippet	35.91	57.26	11.40	38.29	41.40	24.98	10.40	15.91	24.20	38.34
	APO	38.12	57.73	13.57	26.31	46.40	37.77	16.20	19.38	27.48	37.18
	VeriCite	41.63	77.73	10.60	59.09	42.40	45.72	8.40	21.31	25.60	56.41
Gemma2-9B	Vanilla	35.69	77.66	11.27	43.00	40.00	45.92	6.60	15.61	23.20	49.99
	Summary	36.43	72.78	8.80	40.07	41.80	44.30	8.40	15.45	23.21	47.13
	Snippet	34.49	69.60	10.40	41.22	38.60	43.73	7.60	19.26	22.45	47.05
	APO	38.18	50.58	12.13	26.76	49.20	32.79	15.40	16.59	27.35	33.72
	VeriCite	38.93	74.89	9.90	50.66	39.40	63.97	7.40	29.70	23.81	57.16
Glm4-9B	Vanilla	38.58	69.73	14.40	31.24	47.60	40.53	14.40	23.02	27.81	43.80
	Summary	36.43	72.78	11.43	34.61	34.60	22.67	6.00	14.43	22.48	41.43
	Snippet	35.08	66.54	12.27	31.48	30.60	18.18	5.40	8.77	21.55	36.65
	APO	36.83	58.74	11.33	30.35	46.00	41.26	13.80	23.93	25.83	40.24
	VeriCite	43.12	71.30	12.67	39.66	47.00	50.83	12.20	27.41	28.20	49.65
Qwen2.5-7B	Vanilla	37.38	70.99	14.00	42.71	47.60	42.62	12.60	21.75	26.98	48.24
	Summary	37.48	70.32	12.33	39.79	38.40	24.69	7.00	9.57	23.94	41.92
	Snippet	35.38	68.18	12.80	36.97	32.40	19.91	4.80	6.53	22.03	38.95
	APO	36.53	60.69	14.00	24.45	47.40	41.06	14.00	24.45	26.91	38.92
	VeriCite	39.47	76.82	12.13	55.32	49.40	52.87	14.80	38.87	27.70	59.03
Qwen2.5-14B	Vanilla	42.03	69.49	15.67	41.94	53.40	41.73	16.00	20.62	30.60	47.15
	Summary	41.64	63.38	15.10	36.74	45.20	31.24	7.00	8.58	27.37	39.60
	Snippet	39.29	60.83	14.37	36.91	41.00	25.66	6.40	7.43	25.55	37.70
	APO	39.94	56.33	13.87	34.29	54.20	40.17	15.20	22.27	29.32	40.34
	VeriCite	43.50	76.02	13.70	56.90	54.40	50.70	16.20	30.77	30.61	57.57

Table 2: Comparisons between VeriCite and baselines.

	Correct		Citation	
	EM	Recall	Precision	F1
VeriCite	41.63	81.13	74.61	77.73
-w/o init answer	39.24	76.07	71.20	73.55
-w/o evidence selection	38.57	79.42	71.82	75.43
-w/o NLI verification	41.59	70.99	66.95	68.91

Table 3: Ablation study on ASQA.

by the retrieved passages, effectively bypassing the verification process.

The results detailed in Table 3 reveal significant insights. The removal of either the initial answer generation stage or the supporting evidence selection stage induces a substantial decline in answer correctness. In contrast, the detrimental effect on citation quality resulting from these omissions is comparatively less pronounced. This observation indicates that statements originating from both stages possess a complementary nature, collectively contributing to the comprehensiveness of the final answer. Conversely, the ablation of the NLI verification module demonstrates a negligible impact on answer correctness. However, this removal causes a severe deterioration in citation quality. This finding underscores the critical role of the verification step in ensuring the reliability of the citations within the final answer.

	Correct		Citation	
	EM	Recall	Precision	F1
NLI verifier	36.88	84.92	75.71	80.05
Llama3-8B verifier	35.75	76.48	69.85	73.01
DeepSeek-R1 verifier	37.04	82.83	75.92	79.22

Table 4: Results of different verifiers in the VeriCite method.

5.2 Discussion of Verifier

This section discusses the question of verifier model selection within the VeriCite framework. Recognizing that general LLMs possess substantial natural language understanding capabilities, employing the same LLM to perform both answer generation and statement verification tasks within VeriCite offers a promising avenue for significantly reducing framework complexity. Consequently, we investigate an integrated approach where a single LLM is tasked with generating answers and verifying the support for individual statements within retrieved passages. This verification is implemented by instructing the model to output a binary judgment (“Yes” or “No”) regarding whether each statement is supported by its corresponding passage. Furthermore, the experimental design incorporates the current SOTA LLM, DeepSeek-R1, specifically for the verification task. The comparative evaluation utilized the Llama3-8B-Instruct model exclusively for answer generation and evaluated the effectiveness of these three distinct verifier configurations—namely, the

LLM verifier, the DeepSeek-R1 verifier, and the NLI verifier—on a randomly selected subset of 200 samples from the ASQA dataset.

Experimental results present in Table 4. Utilizing a general LLM for dual-role verification proved detrimental, leading to a noticeable decline in both answer correctness and citation quality relative to the NLI verifier. In contrast, the DeepSeek-R1 verifier achieved a marginal improvement in answer correctness compared to the NLI verifier, while its impact on citation quality was nearly equivalent. In terms of computational efficiency, while the LLM and NLI verifiers are comparable in both parameter sizes and operational costs, their practical deployment costs are substantially lower than those of the large-scale DeepSeek-R1 model.

Therefore, based on this empirical evaluation balancing performance gains against resource expenditure, selecting the NLI model for the verification role emerges as the optimal choice, offering an effective and cost-efficient solution.

6 Conclusion

In this paper, we propose VeriCite, a novel framework designed to enhance citation quality in RAG systems. The framework operates through three sequential stages: initial answer generation, supporting evidence selection, and final answer refinement. Experimental results demonstrate that VeriCite significantly enhances citation quality while maintaining answer correctness comparable to the strongest baseline methods. Furthermore, ablation studies confirm the necessity of each core component within the framework. Additionally, the paper discusses the critical importance of selecting a NLI model for the verification role, providing justification for this design choice.

Acknowledgments

This work was funded by the National Natural Science Foundation of China (NSFC) under Grants No. 62372431, 62472408 and 62441229, the Strategic Priority Research Program of the CAS under Grants No. XDB0680301, the National Key Research and Development Program of China under Grants No. 2023YFA1011602, the Lenovo-CAS Joint Lab Youth Scientist Project, and the project under Grants No. JCKY2022130C039. This work is also supported by Meituan and Baidu Inc. All content represents the opinion of the authors, which is not necessarily shared or endorsed by their respective employers and/or sponsors.

References

- [1] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems* 33 (2020), 1877–1901.
- [2] Ido Dagan, Dan Roth, Mark Sammons, and Fabio Massimo Zanzotto. 2015. Recognizing Textual Entailment: Models and Applications. *Computational Linguistics* 41, 1 (2015).
- [3] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783* (2024).
- [4] 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*. 3558–3567.
- [5] Yixing Fan, Qiang Yan, Wenshan Wang, Jiafeng Guo, Ruqing Zhang, and Xueqi Cheng. 2025. TrustRAG: An Information Assistant with Retrieval Augmented Generation. *arXiv preprint arXiv:2502.13719* (2025).
- [6] Katja Filippova. 2020. Controlled Hallucinations: Learning to Generate Faithfully from Noisy Data. In *Findings of the Association for Computational Linguistics: EMNLP 2020*. 864–870.
- [7] Luyu Gao, Zhu Yun Dai, Panupong Pasupat, Anthony Chen, Arun Tejasvi Chaganty, Yicheng Fan, Vincent Zhao, Ni Lao, Hongrae Lee, Da-Cheng Juan, et al. 2023. 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)*. 16477–16508.
- [8] 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*. 6465–6488.
- [9] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997* (2023).
- [10] Team GLM, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Dan Zhang, Diego Rojas, Guanyu Feng, Hanlin Zhao, et al. 2024. Chatglm: A family of large language models from glm-130b to glm-4 all tools. *arXiv preprint arXiv:2406.12793* (2024).
- [11] Jiafeng Guo, Yixing Fan, Xiang Ji, and Xueqi Cheng. 2019. MatchZoo: A Learning, Practicing, and Developing System for Neural Text Matching. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (Paris, France) (SIGIR'19)*. Association for Computing Machinery, New York, NY, USA, 1297–1300. doi:10.1145/3331184.3331403
- [12] Or Honovich, Roei Aharoni, Jonathan Herzig, Hagai Taitelbaum, Doron Kukliansky, Vered Cohen, Thomas Scialom, Idan Szpektor, Avinatan Hassidim, and Yossi Matias. 2022. TRUE: Re-evaluating Factual Consistency Evaluation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 3905–3920.
- [13] Nan Hu, Jiaoyan Chen, Yike Wu, Guilin Qi, Sheng Bi, Tongtong Wu, and Jeff Z Pan. 2024. Benchmarking large language models in complex question answering attribution using knowledge graphs. *arXiv preprint arXiv:2401.14640* (2024).
- [14] Jie Huang and Kevin Chen-Chuan Chang. 2023. Citation: A key to building responsible and accountable large language models. *arXiv preprint arXiv:2307.02185* (2023).
- [15] 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. *Comput. Surveys* 55, 12 (2023), 1–38.
- [16] 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 (2020), 9459–9474.
- [17] Dongfang Li, Zetian Sun, Baotian Hu, Zhenyu Liu, Xinshuo Hu, Xuebo Liu, and Min Zhang. 2024. Improving Attributed Text Generation of Large Language Models via Preference Learning. *arXiv preprint arXiv:2403.18381* (2024).
- [18] Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*. 74–81.
- [19] Aixian Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Cheng-gang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437* (2024).
- [20] Xiao Liu, Hanyu Lai, Hao Yu, Yifan Xu, Aohan Zeng, Zhengxiao Du, Peng Zhang, Yuxiao Dong, and Jie Tang. 2023. WebGLM: Towards an efficient web-enhanced question answering system with human preferences. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 4549–4560.
- [21] Ziwei Liu, Liang Zhang, Qian Li, Jianghua Wu, and Guangxu Zhu. 2024. Invar-RAG: Invariant LLM-aligned Retrieval for Better Generation. *arXiv preprint arXiv:2411.07021* (2024).
- [22] Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2020. AmbigQA: Answering ambiguous open-domain questions. *arXiv preprint arXiv:2004.10645* (2020).
- [23] Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. 2021. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332* (2021).
- [24] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*. 311–318.
- [25] Ori Press, Andreas Hochlehnert, Ameya Prabhu, Vishal Udandarao, Ofir Press, and Matthias Bethge. 2024. CiteME: Can Language Models Accurately Cite Scientific Claims? In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.
- [26] Zhen Qin, Rolf Jagerman, Kai Hui, Honglei Zhuang, Junru Wu, Le Yan, Jiaming Shen, Tianqi Liu, Jialu Liu, Donald Metzler, et al. 2024. Large Language Models are Effective Text Rankers with Pairwise Ranking Prompting. In *Findings of the Association for Computational Linguistics: NAACL 2024*. 1504–1518.
- [27] 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* 49, 4 (2023), 777–840.

- [28] Devendra Sachan, Mike Lewis, Mandar Joshi, Armen Aghajanyan, Wen-tau Yih, Joelle Pineau, and Luke Zettlemoyer. 2022. Improving Passage Retrieval with Zero-Shot Question Generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. 3781–3797.
- [29] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* (2017).
- [30] Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. 2021. Retrieval Augmentation Reduces Hallucination in Conversation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*. 3784–3803.
- [31] 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*. 8273–8288.
- [32] Hao Sun, Hengyi Cai, Bo Wang, Yingyan Hou, Xiaochi Wei, Shuaiqiang Wang, Yan Zhang, and Dawei Yin. 2024. Towards Verifiable Text Generation with Evolving Memory and Self-Reflection. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. 8211–8227.
- [33] Weiwei Sun, Lingyong Yan, Xinyu Ma, Shuaiqiang Wang, Pengjie Ren, Zhumin Chen, Dawei Yin, and Zhaochun Ren. 2023. Is ChatGPT Good at Search? Investigating Large Language Models as Re-Ranking Agents. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. 14918–14937.
- [34] Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118* (2024).
- [35] Nandan Thakur, Luiz Bonifacio, Xinyu Zhang, Odunayo Ogundepo, Ehsan Kamaloo, David Alfonso-Hermelo, Xiaoguang Li, Qun Liu, Boxing Chen, Mehdi Rezagholizadeh, et al. 2023. NoMIRACL: Knowing When You Don’t Know for Robust Multilingual Retrieval-Augmented Generation. *arXiv preprint arXiv:2312.11361* (2023).
- [36] Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. MuSiQue: Multi-hop Questions via Single-hop Question Composition. *Transactions of the Association for Computational Linguistics* 10 (2022), 539–554.
- [37] Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2023. Interleaving Retrieval with Chain-of-Thought Reasoning for Knowledge-Intensive Multi-Step Questions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 10014–10037.
- [38] Long Xia, Jun Xu, Yanyan Lan, Jiafeng Guo, and Xueqi Cheng. 2015. Learning Maximal Marginal Relevance Model via Directly Optimizing Diversity Evaluation Measures. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval (Santiago, Chile) (SIGIR ’15)*. Association for Computing Machinery, New York, NY, USA, 113–122. doi:10.1145/2766462.2767710
- [39] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. 2024. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115* (2024).
- [40] 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*. 2369–2380.
- [41] Wenpeng Yin, Dragomir Radev, and Caiming Xiong. 2021. DocNLI: A Large-scale Dataset for Document-level Natural Language Inference. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. 4913–4922.
- [42] Jiajie Zhang, Yushi Bai, Xin Lv, Wanjun Gu, Danqing Liu, Minhao Zou, Shulin Cao, Lei Hou, Yuxiao Dong, Ling Feng, and Juanzi Li. 2024. LongCite: Enabling LLMs to Generate Fine-grained Citations in Long-context QA. *arXiv preprint arXiv:2409.02897* (2024).

A Prompt

A.1 Initial Answer Generation

Prompt Template for Initial Answer Generation

Instruction: Please refer to the information in the following passages to answer the question. When answering, ignore any irrelevant information from the passages, but retain all relevant details to provide a comprehensive and accurate response. Always cite for any factual claim. When citing several search results, use [1][2][3]. Cite at least one passage in each sentence.

Question: {Question}

Document: [1](Title: {Title}): {Passage}

Document: [2](Title: {Title}): {Passage}

...

Answer:

A.2 Supporting Evidence Check

Prompt Template for Supporting Evidence Check

Instruction: Please refer to the information in the following passage to answer the question. You need to first determine whether the information in the passage is helpful for answering the question. If you believe the passage is helpful, output 'Yes'; otherwise, output 'No'. Do not output any additional content.

Question: {Question}

Passage: {Passage}

Response:

A.3 Supporting Evidence Extraction

Prompt Template for Supporting Evidence Extraction

Instruction: Please refer to the information in the following passage to answer the question. When answering, ignore any irrelevant information from the passage, but retain all relevant details to provide a comprehensive and accurate response.

Question: {Question}

Passage: {Passage}

Response:

A.4 Final Answer Refinement

Prompt Template for Final Answer Refinement

Instruction: Please answer the following question. I will provide you with some answer statements with citations, as well as their original references. You need to summarize these statements and merge their citations such as [1][2].

Question: {Question}

References:

Document: [1](Title: {Title}): {Passage}

Document: [2](Title: {Title}): {Passage}

...

Answer statements:

{Statement 1} [citation ids]

{Statement 2} [citation ids]

...

Your Answer: