CaLM: Contrasting Large and Small Language Models to Verify Grounded Generation

I-Hung Hsu[1]*, **Zifeng Wang**[2], **Long T. Le**[2], **Lesly Miculicich**[2], **Nanyun Peng**[3], **Chen-Yu Lee**[2], **Tomas Pfister**[2]
[1]University of Southern California, [2]Google Cloud AI Research, [3]University of California, Los Angeles

Abstract

Grounded generation aims to equip language models (LMs) with the ability to produce more credible and accountable responses by accurately citing verifiable sources. However, existing methods, by either feeding LMs with raw or preprocessed materials, remain prone to errors. To address this, we introduce CaLM, a novel verification framework. CaLM leverages the insight that a robust grounded response should be consistent with information derived solely from its cited sources. Our framework empowers smaller LMs, which rely less on parametric memory and excel at processing relevant information given a query, to validate the output of larger LMs. Larger LM responses that closely align with the smaller LMs' output, which relies exclusively on cited documents, are verified. Responses showing discrepancies are iteratively refined through a feedback loop. Experiments on three open-domain question-answering datasets demonstrate significant performance gains of 1.5% to 7% absolute average without any required model fine-tuning.

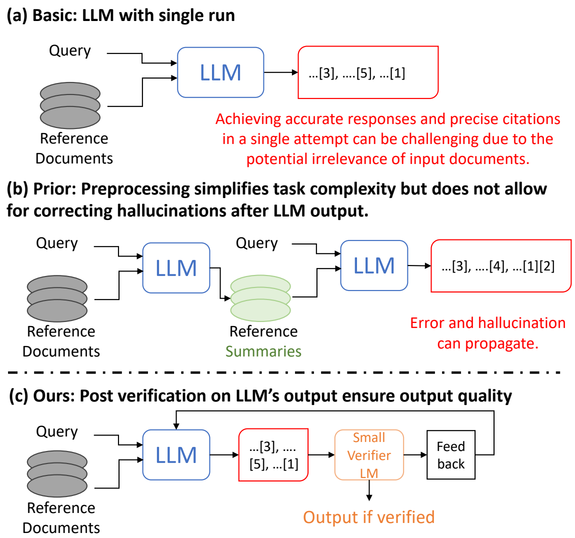

Figure 1: Comparison between different categories of existing inference methods for grounded generation. (a) LLM with single-run can hallucinate easily due to the high complexity of the task. (b) Preprocessing methods reduce task complexity but the hallucination issues can propagate from preprocessing steps. (c) We propose using verification and rectification to ensure LLMs generate outputs with complete citations and accurate answers, maintaining quality.

1 Introduction

Large Language Models (LLMs) are increasingly popular tools for information seeking. A recent trend emphasizes integrating citations from verifiable sources to boost LLM credibility and enable user verification. This aims to reduce hallucinations and increase accountability (Gao et al., 2023b; Huang and Chang, 2023; Liu et al., 2023b). To achieve this, LLMs must not only identify relevant documents within vast retrieved collections but also accurately ground their responses in these sources and effectively generate their responses. This significantly increases the complexity of LLM operations (Gao et al., 2023b).

A standard approach to achieve this grounded generation is by retrieval-augmented generation with instructions to guide LLMs to generate responses along with their corresponding sources in one single LLM inference run (see Fig. 1 (a)). More recently, more sophisticated approaches utilize LLMs to first summarize relevant documents (Gao et al., 2023b) or use key information extraction and algorithms that explore different relevant document combinations by asking LLMs to enrich

* Work done while the author was a student researcher at Google Cloud AI Research. Correspondence to: I-Hung Hsu <ihunghsu@usc.edu>, Chen-Yu Lee <chenyulee@google.com>

Findings of the Association for Computational Linguistics ACL 2024, pages 12782–12803
August 11-16, 2024

the original input query with additional information (Li et al., 2023) (see Fig. 1 (b)).

However, both single-run (Fig. 1 (a)) and preprocessing (Fig. 1 (b)) strategies face challenges for accurate generation and citation. Single-run approaches require LLMs to process the input query and a potential large volume of retrieved documents in one forward pass, which can strain their capabilities. Preprocessing approaches, while more focused, risk error propagation or loss of information. Additionally, both strategies limit the LLM's ability to iterate, refine, and verify responses, impacting citation accuracy and answer correctness.

In contrast to single-run and preprocessing strategies, we propose a novel post-verification approach that enables LLMs to fact-check and ground their responses. Our design leverages the complementary strengths of larger and smaller LMs. We observe that larger LMs excel at identifying relevant information within a vast corpus but can rely excessively on internal parametric memory during generation. Smaller LMs, however, are adept at processing retrieved relevant information but less capable of identifying it from large collections (see § 3.3 for details).

Building on these observations, we propose CaLM (**C**ontr**a**sting **L**arge and s**M**all language models to verify grounded generation). CaLM validates the large LLM's response by cross-referencing it with output from a smaller LM. The smaller LM scrutinizes the cited documents to confirm the large LLM's citation accuracy. If the responses align, the large LLM's answer is verified. If not, CaLM extracts useful statements and evidence from the large LLM's response and seeks additional supporting information to improve the query response. Importantly, CaLM requires no model fine-tuning, allowing smaller LMs to significantly enhance the grounded generation capabilities of large LMs. Fig. 3 illustrates this process.

We conduct experiments on three open-domain question answering datasets (QAMPARI, ASQA, and ELI5), which require consulting multiple sources for comprehensive answers. Our method demonstrates significant improvements in both answer accuracy and citation quality, outperforming state-of-the-art methods by an average of 1.5% to 7%. Crucially, our method remains robust even in challenging scenarios with less powerful retrieval systems, while other baselines often struggle.

2 Problem Statement

Task Setup. We cope with the problem of grounded generation (Gao et al., 2023b). Given a query q and a corpus of trustworthy text passages $\mathcal{D}$, the model needs to generate an answer response $\mathcal{A}$, which consists of n statements $s_1, s_2, ...s_n$, based on the knowledge in $\mathcal{D}$. Each statement s_i cites a list of passages $\mathcal{C}_i = \{c_i^1, c_i^2, ...\}, \forall c_i^j \in \mathcal{D}$. The collective sets $\mathcal{C}_i$ for $i = 1, 2, \ldots, n$ constitute the grounded evidence $\mathcal{G}$, from which $\mathcal{A}$ is derived. Our goal is to jointly optimize the usefulness of $\mathcal{A}$ to q, the preciseness of $\mathcal{C}_i$ for statement s_i, and the integrity of $\mathcal{G}$ to adequately support $\mathcal{A}$.

Evaluation of Response. The task involves measuring three dimensions of system responses, following the setup from Gao et al. (2023b).

- **Fluency**: Determining whether the model's generated text $\mathcal{A}$ is fluent and coherent.
- **Correctness**: Assessing if $\mathcal{A}$ is accurate and covers **all relevant aspects** of query q.
- **Citation Quality**: Evaluating whether cited passages directly support the answer and avoids irrelevant citations. This is achieved by evaluating both **citation recall** and **citation precision**. Gao et al. (2023b) propose measuring citation quality by averaging scores for each statement s_i. Citation recall ensures there is at least one supporting citation c_i^j for s_i. Citation precision measures whether all the citations are "relevant". Specifically, a citation c_i^j is considered "irrelevant" to statement s_i if c_i^j cannot support s_i, and removing c_i^j from $\mathcal{C}_i$ would not impact the overall support for s_i from the remaining citations.

3 Automated Verification for Grounded Generation

Although LLMs have demonstrated proficiency in a wide range of tasks, they remain susceptible to generating hallucinations (Huang et al., 2023; Zhang et al., 2023). These hallucinations could occur in both answers and citations within grounded generations due to the high complexity of the entire working pipeline, which includes noise from the retrievers and the limited ability of LLMs to handle long contexts (Liu et al., 2023a). This issue underscores the critical need for verification mechanisms to ensure the quality of the generated output and to leverage the interplay between verification and generation systems to improve the final output $\mathcal{A}$.

In this section, we first analyzes key factors to verify grounded generation (§ 3.1). Subsequently, we introduce an automated and unsupervised verification method for grounded generation using a small LM as a verifier and contrasting results from large and small LMs to verify large LMs' response (§ 3.2 & 3.3).

3.1 Key Factors for Automated Verification

Automated verification, unlike the task evaluation in § 2, operates without a ground-truth reference and should be efficient for real-time system feedback. Here, our focus lies on assessing answer correctness and citation quality. To evaluate the **correctness** of a generated grounded response, we must ensure that generated responses ($\mathcal{A}$) faithfully leverage information from the knowledge base $\mathcal{D}$, avoiding hallucinations or model biases. Additionally, *correct reasoning* in deriving the answer is also crucial. For the automatic evaluation of **citation quality**, a trained Natural Language Inference (NLI) model can assess each citation and statement pair iteratively to measure the citation's fidelity (Gao et al., 2023b). However, this process can be computationally expensive for lengthy generated answers with numerous citations. Efficient automated verification for grounded generation must consider these factors.

3.2 Contrasting Large and Small LMs for Automated Verification

We propose a verification method using a smaller LM to assess the quality of a larger LM's grounded generation. The small LM receives only the large LM's *cited documents ($\mathcal{G}$)* as input to answer the same query q. Consistency between their responses indicates the quality of $\mathcal{G}$ and the grounded generation from the large LM.

Our design exploit the inherent characteristics of smaller LMs. We posit that a robust $\mathcal{G}$ should enable even small LMs to deduce the correct answer. Notably, small LMs, having fewer parameters, are demonstrably more receptive to integrating external knowledge (Xie et al., 2023). Reaching consistent results from both LMs indicate high answer fidelity. [1] Leveraging different LMs as support also *reduce the reasoning error risks* as the different LMs exhibit diverse strengths and reasoning mechanisms (Jiang et al., 2023). These characteristics of small LMs makes our design effective for verifying the answer correctness.

Furthermore, as will detailed in § 3.3, smaller LMs are more sensitive to the relevance of input evidence. Irrelevant documents in the evidence set $\mathcal{G}$ can easily mislead small LMs, while missing crucial citations hinder their ability to reach the correct answer independently, due to their limited parametric knowledge. This sensitivity allows us to utilize small LMs for assessing the quality of $\mathcal{G}$.

3.3 Analyzing Model Size Impact on LMs' Sensitivity to Input Document Relevance

Our automated verification method exploits the high sensitivity of small LMs. This section empirically demonstrates that, within the same model family, smaller LMs exhibit greater sensitivity to the relevance of input documents to a given query q compared to their larger counterparts.

We investigate the sensitivity of LMs to the relevance of input documents by modeling the performance of an LM as a function of the input document's relevance. To mitigate the effects of the divergent abilities for different LMs to follow instructions, we examine the *relative* performance improvement of the model response as the relevance of the input document increases. The larger this value is, the more sensitive the LM is.

Yet, the complexity of this function is beyond simple linearity. As depicted in Fig. 2(a), this function typically exhibits a monotonic increase and hence we intend to additionally study second-order improvements gain analysis to further study the curvature of the function. This analysis use three anchoring points $x_{low}, x_{med}, x_{high}$, and the corresponding performance $P(x_{low}), P(x_{med}), P(x_{high})$ to study the much incremental performance gain the LM can get when the input document's relevance keep increasing. Specifically, we use the ratio $\frac{P(x_{high})-P(x_{med})}{P(x_{med})-P(x_{low})}$ to represent the incremental performance gain. A higher ratio indicates the LM is more sensitive to input document's relevance, as illustrated by the orange line in Fig. 2(a).

We conduct empirical studies using a retrieval-augmented generation setting on the ASQA dataset (Stelmakh et al., 2022). Each instance in the dataset contains multiple answers and requires reading multiple documents to make correct prediction. In this experiment, a LM utilizes five input documents $\mathbf{d}$ to answer queries q, with model per-

[1] In this paper, we differentiate LMs by size, labeling them as large or small. However, a more accurate categorization would be strong versus weak LMs, reflecting their varying performance levels across different LM families.

Fig. (a): The illustration of describing the accuracy of LM as a function of the input document relevancy to the query

Fig. (b): Relative Improvement in QA accuracy when input high relevant documents compared to input noisy ones.

Fig. (c): 2nd Order Relative Improvement Analysis in QA accuracy when adjusting input document relevancy.

Figure 2: The studies of the performance of an LM as a function of the input document's relevance score using the ASQA dataset. We show that, *within the same LM family, smaller LMs demonstrate higher sensitivity to the relevance of the input document, when anchored to the largest model in the family*. ***(a)*** The illustration of the function. This function is a monotonic increase function as the accuracy always increase when input document's relevance score increase. Hence, studying the second order relative improvements can help us know the *incremental performance gain* for the LM when the input document's relevance keep increasing. ***(b)*** The result of relative improvement. ***(c)*** The result of the second order relative improvement analysis. From (b)(c), we can observe that smaller models tend to exhibit greater relative improvements and achieve larger incremental performance gains compared to their larger counterparts.

formance evaluated based on answer accuracy, and the relevance of the input documents $\mathbf{d}$ is assessed by their average recall rate in containing answers. We prepare three different set of $\mathbf{d}$ with approximately 27%, 56%, and 78% recall rates, respectively, to represents $x_{low}, x_{med}, x_{high}$ in Fig. 2(a).

Fig. 2(b) presents the results of relative performance improvement, while Fig. 2(c) shows the second-order analysis. Our study encompasses LM families such as Yi-LM, Vicuna (Zheng et al., 2023), WizardLM (Xu et al., 2023), and Tulu-2-dpo (Ivison et al., 2023). The findings indicate that within the same LM family, compared to the largest model, smaller LMs usually achieve higher relative improvement and receive a greater incremental performance gain, suggesting *their higher sensitivity to the relevance of input documents to the query* q. On the other hand, smaller LMs are more prone to erroneous results and suboptimal performance with irrelevant documents, which motivates our automated verification method design.

4 CaLM Framework

Building on the automated verification design from § 3.2, we introduce CaLM, an inference framework that leverages the synergy between verification and generation systems to enhance the system's final grounded generation.

Fig. 3 depicts the iterative five-step algorithm of our CaLM framework. Steps three and four correspond to the automated verification method detailed in § 3.2. The remaining steps involve the large LM making predictions.

We now present a detailed breakdown of each step below. We differentiate between the "main LM," whose responses are verified, and the "verifier LM," an auxiliary LM assisting the verification process. As described in § 3.2, we recommend using a larger LM as the main LM and a smaller LM for verification to achieve optimal performance.

Step (1) Context Retrieval. CaLM starts by selecting a *ranked pool* of trustworthy passages p for a given query q using a retriever.

Step (2) Main LM Generation. We select the top-k documents from the reference pool p and feed them into the main LM. This value of k is a hyperparameter constrained by the maximum input capacity of the main LM. The LM then analyzes these k passages to generate an answer candidate $\mathcal{A}$ for the query q. Our findings in § 3.3 suggest employing a large-scale LM at this stage. This is because larger LMs exhibit greater robustness in accurately identifying useful information and filtering out noise within the retrieved passages.

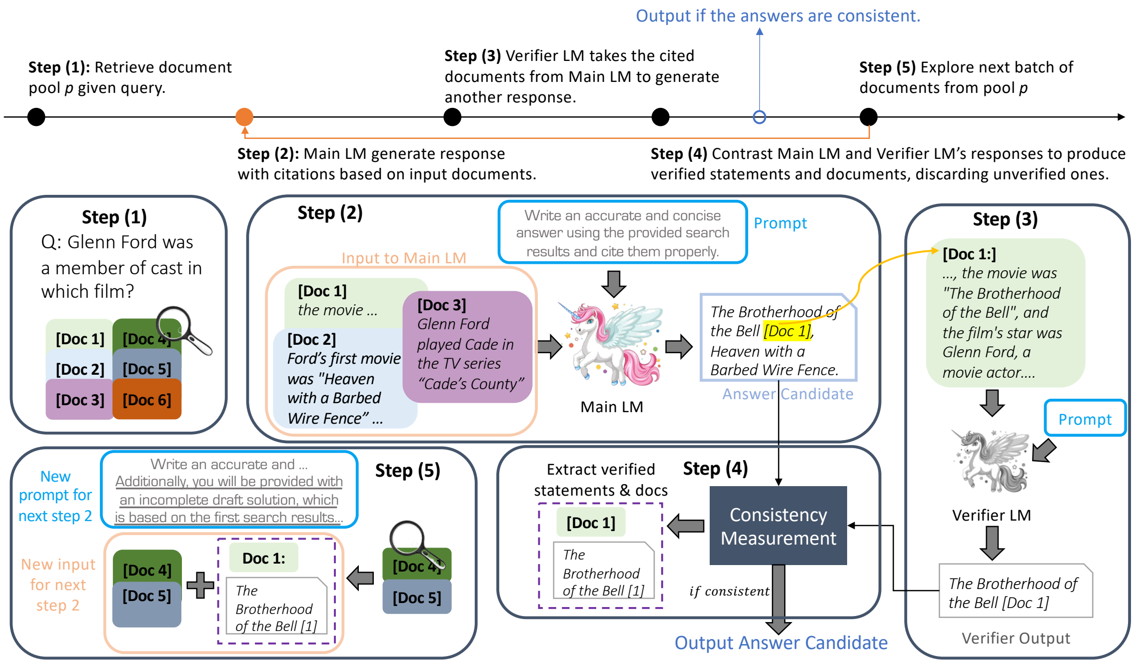

Figure 3: Overview of CaLM: ***Top:*** The flow diagram of our method. ***Bottom:*** A detailed depiction of each step's operation. The algorithm starts with a retriever extract a relevant document pool p for the input query (Step (1)). Then, the main language model (LM) takes the first batch of documents and employs retrieval-augmented generation to produce an answer candidate, which cites relevant supporting documents (Step (2)). Subsequently, this candidate is validated by contrasting it with the verifier output from the verifier LM (Steps (3) & (4)). Our verifier LM evaluates citation quality by accessing only the documents cited by the main LM's response, rather than the same input documents. For responses with sufficient consistency, we accept the answer candidate directly. If inconsistent, we break down the answer candidate into individual statements, retaining only those corroborated by similar arguments in the verifier output for further correction in next iteration (Step (5)).

Step (3) Verifier LM Generation. Building upon the automated verification design outlined in § 3.2, this step leverages a grounded evidence set $\mathcal{G}$ extracted from the main LM's answer candidate $\mathcal{A}$. A smaller, dedicated verification LM then re-attempt the grounded generation process and obtain verifier output $\mathcal{A}'$ for query q by solely providing the model with the evidence set, $\mathcal{G}$, rather than the full top-k documents.

Step (4) Contrasting Answer Candidate and Verifier Output: A strong evidence $\mathcal{G}$ should enable small LMs to deduce correct answers reliably. Our goal in this step is to verify consistency (Wang et al., 2023) of $\mathcal{A}'$ against the answer candidate $\mathcal{A}$.

If the comparison shows enough consistency, we accept the answer $\mathcal{A}$ and stop the iterative process. Otherwise, we dismiss the inconsistent segments from the answer and citations, and continue with the next iteration. More technically, we extract $\bar{\mathcal{A}} = \mathcal{A} \cap \mathcal{A}'$ and the corresponding $\bar{\mathcal{G}}$ within $\bar{\mathcal{A}}$ for Step (5) usage.

The realization of consistency measurement is done via calculating ROUGE-2 score between the $\mathcal{A}'$ and $\mathcal{A}$. If the ROUGE-2 score exceeds a threshold θ, the answer candidate is considered acceptable. This threshold can be tuned using a small development set. Empirically, by our observation, setting $\theta = 0.2$ to 0.5 yields satisfactory results by small set of qualitative examination.

Step (5) Input Preparation for Next Iteration: In this final step, we prepare input for the next iteration, including input reference document lists and the draft for correction. The new input reference document lists is initialized by $\bar{\mathcal{G}}$ and is supplemented more passages from the next batch of passages from pool p until complete the budget k. This process boosts the likelihood of finding relevant documents while maintaining useful documents that have been verified. Then, we create a new prompt for large LM focusing on leveraging the new input reference document lists to correct the incomplete answer response $\bar{\mathcal{A}}$.

The full prompt we use for each step are detailed in Appx. §B. In practice, we should set a maximum iteration T to halt the whole process to prevent the verification condition cannot be satisfied. If this

Dataset	Example
ASQA	**Question:** Who sings don't tell me what to do? **Reference Answer:** Marty Stuart recorded the song Don't Tell Me What to Do, recorded as I'll Love You Forever (If I Want To) in 1988. Pam Tillis sang Don't Tell Me What To Do in 1990, and in 1993, the Baby Animals recorded the song.
QAMPARI	**Question:** Which movie did John Carpenter direct for which he also composed the music? **Reference Answer:** Vampires, In the Mouth of Madness, Assault on Precinct 13, Dark Star, Big Trouble in Little China, They Live, Halloween, Escape from New York, Prince of Darkness, Ghosts of Mars, The Fog, Chevil, Village of the Damned
ELI5	**Question:** Why are fruit in Chinese supermarkets so much bigger than western chains like loblaws or food basics? **Reference Answer:** There are a lot less restrictions on using chemicals in agriculture here, so some farmers use 'growth accelerators' to make their fruit (and other products) bigger. Last May, overuse of the chemicals resulted in a spate of [exploding watermelons]

Table 1: Illustrative examples from each experiment's dataset.

maximum iteration is reached, we will output the last answer candidate we get from the main LM.

5 Experimental Setup

We consider three factoid question-answering (QA) datasets. For each dataset, we present an illustrative example in Tab. 1 for better understanding.

We prepare the trustworthy text passages $\mathcal{D}$ for each dataset accordingly following (Gao et al., 2023b). Each entry in $\mathcal{D}$ is a 100-word passage following previous works on open-domain QA (Karpukhin et al., 2020; Petroni et al., 2021; Liu et al., 2023a).

5.1 The ASQA dataset

Basic Introduction. ASQA (Stelmakh et al., 2022) is a long-form generation QA dataset derived from the AmbigQA dataset (Min et al., 2020). It comprises questions characterized by their ambiguity, necessitating multiple short answers to address various aspects. Each entry in the dataset is accompanied by a comprehensive long-form answer that covers all the corresponding short answers. 948 samples are tested for our experiment.

Experimental Setting Since most questions can be answered by Wikipedia, prior works usually use 2018-12-20 Wikipedia snapshot as $\mathcal{D}$.

For retriever utilization, we examine the application of both DPR (Karpukhin et al., 2020) and GTR-large (Ni et al., 2022). DPR introduces marginally more complex challenges for the LLM, achieving a recall rate of 51.5%, whereas GTR achieves 56.8% when considering the top-5 retrieved documents.

Evaluation

- **Fluency**: We use **MAUVE** score (Pillutla et al., 2021) to evaluate the corpus-wise similarity of the machine generated text and the long answers generated by systems.
- **Correctness**: We follow (Stelmakh et al., 2022) to calculate the exact matching recall (**EM recall**) of the presents of correct short answers.
- **Citation quality**: As mentioned in § 2, we calculate **citation recall** and **citation precision** using the scripts provided by (Gao et al., 2023b).

5.2 The QAMPARI dataset

Basic Introduction. QAMPARI (Amouyal et al., 2022) is created from Wikipedia, pairing questions with multiple answers derived from its knowledge graph and tables. These answers, comprised of entities, describe simple relationships to the entities in the query q. As a result, this dataset focuses on testing systems' abilities on entity identification within questions and accurately pinpointing the relevant entities. We use the same 1000 testing samples used in (Gao et al., 2023b) for experiments.

Experimental Setting Similar to the ASQA case, we employ the Wikipedia snapshot from 2018-12-20, as our $\mathcal{D}$. For retrieval, we again use both DPR and GTR-large, achieving recall rates of approximately 17.6% and 31.6%, respectively, for the top five retrieved documents for each query.

Evaluation Metrics In QAMPARI, we only considering the correctness and the citation quality since the output of the dataset is a list of entities.

- **Correctness**: We follow (Stelmakh et al., 2022) to to calculate the entity **precision** and recall of the model prediction using *exact string match*. When calculate recall, the evaluation considers recall to be 100% if the prediction includes at least 5 correct answer, denoted as **recall-5**.
- **Citation quality**: We use the same way as the

ASQA dataset to evaluate the citation quality.

5.3 The ELI5 dataset

Basic Introduction. The ELI5 dataset, introduced by (Fan et al.), primarily features "How," "Why," and 'What" questions. It tests a system's ability to summarize complex information into clear and insightful answers. We use the 1000 samples used in (Gao et al., 2023b) for our experiment.

Experimental Setting Unlike ASQA and QAMPARI, the ELI5 dataset covers diverse topics and hence, documents in Sphere corpus are treated as $\mathcal{D}$ (Piktus et al., 2021). Given the large size of the Sphere corpus, BM25 is used for efficient retrieval.

Evaluation Metrics

- **Correctness**: ELI5 dataset does not provide short entity answers. We follow (Gao et al., 2023b) to calculate **claim recall** for correctness. For each reference answer in the dataset, three "sub-claims" are first extracted, and we will test whether the machine's answer $\mathcal{A}$ can entail these sub-claims using a TRUE NLI model (Honovich et al., 2022)
- **Fluency** & **Citation quality**: We use the metrics as the ASQA dataset for evaluation.

5.4 Compared Methods

We compare the following methods, all of which we have independently rerun, except Self-RAG. For our own rerun results, the reported results are the average of three random runs.

1. **In-Context Learning (ICLCite)**: LLMs are invoked once for grounded generation through instruction-based in-context learning. Following (Gao et al., 2023b), we provide five documents to the LLM. They suggest that increasing the number of input document lists does not significantly improve performance when using GPT-3.5-turbo.
2. **Summary then In-Context Learning (Summ + ICLCite)** (Gao et al., 2023b): This method follows the preprocessing paradigm shown in Fig. 1. Initially, LLMs generate summaries for each document based on the query q. Then, these summaries are fed into the LLM to execute ICLCite. In our experiments, we generate summaries for the top-9 documents retrieved for each instance.
3. **Snippet then In-Context Learning (Snippet + ICLCite)** (Gao et al., 2023b): Similar to Summ + ICLCite, but LLM are guided to generate extractive summaries during preprocessing steps.
4. **In-Context Learning with Self-Consistency (ICLCite + USC)**: This post-processing method that employs ICLCite to initially generate various output samples. Then, it applies universal self-consistency Chen et al. (2023) to obtain the results. For a fair comparison with other baselines, like Summ+ICLCite, we first generate 9 samples and then use a LLM to determine the most consistent result among them.
5. **Self-RAG** (Asai et al., 2023): This method finetunes LMs to generate special tokens to trigger additional fact checks and retrieval. Since this method requires model finetuning, we only report results on ASQA dataset only.
6. **CaLM**: We use verification and an iterative refinement design to ensure the output quality. We follow the setting of ICLCite to set the our main LM's reading budget $k = 5$. We set the consistency threshold $\theta = 0.25$ for the ELI5 dataset and $\theta = 0.5$ for the ASQA dataset. This θ is decided by manual qualitative examination on a small set of development data. Besides, we set the maximum iteration to be 4 for budget concern, and use the 13B version of tulu-2-dpo (Ivison et al., 2023) as the verifier LM.

6 Experimental Results

6.1 Main Results

For the main experiment, we consider two different large LMs as the backbone: GPT-3.5-Turbo-1106 (Ouyang et al., 2022) and the PaLM-based LLMs (Anil et al., 2023), text-unicorn [2].

Tab. 2, Tab. 3, and Tab. 4 present the results on ASQA, QAMPARI, and ELI5, respectively. We have three discovery across the three datasets:

1. Snippet+ICLCite performs as the strongest baseline. We hypothesize that for tasks involving grounded generation, preserving original evidence within documents is crucial for citation quality. Abstractive summarization can result in the loss of significant evidence crucial for solving the task. Additionally, accurately extracting consistent answers from multiple samples enhances correctness, yet determining corresponding consistent citations poses a challenge.
2. Despite Snippet+ICLCite being the strongest baseline, it does not always outperform ICLCite.

[2] https://cloud.google.com/vertex-ai/docs/generative-ai/model-reference/text

Method	Method Type	# Main LM Call	# Verifier LM Call	DPR as retriever: Fluency mauve	Correct. EM Rec.	Citation Prec.	Citation Rec.	Average	GTR as retriever: Fluency mauve	Correct. EM Rec.	Citation Prec.	Citation Rec.	Average
GPT-3.5-Turbo-1106 as Main LM													
ICLCite (Gao et al., 2023b)	Single Run	1	-	74.73	39.32	**67.36**	69.48	62.72	71.85	41.92	**73.14**	77.90	66.20
Summ + ICLCite (Gao et al., 2023b)	Preprocess	10	-	48.95	29.30	60.14	54.52	48.23 (-14.49)	68.01	41.11	66.04	74.43	62.40 (-3.8)
Snippet + ICLCite (Gao et al., 2023b)		10	-	48.56	29.48	59.52	53.84	47.85 (-14.87)	68.84	39.89	62.05	71.06	60.46 (-5.74)
ICLCite + USC (Chen et al., 2023)*	Postprocess	10	-	77.50	40.71	61.20	64.07	60.87 (-1.85)	77.31	42.75	67.08	71.64	64.69 (-1.51)
CaLM (ours)		≤ 4	≤ 3	**81.35**	**43.56**	66.00	**69.95**	**64.71** (+1.99)	**83.98**	**45.01**	72.59	**78.03**	**68.98** (+2.78)
text-unicorn as Main LM													
ICLCite (Gao et al., 2023b)	Single Run	1	-	62.01	37.09	62.42	60.35	55.46	63.25	39.83	69.39	67.98	60.11
Summ + ICLCite (Gao et al., 2023b)	Preprocess	10	-	63.21	38.67	52.41	59.45	53.43 (-2.03)	75.68	42.65	61.18	68.91	62.11 (+2.00)
Snippet + ICLCite (Gao et al., 2023b)		10	-	59.03	37.69	54.62	59.44	52.69 (-2.77)	72.50	40.97	60.88	68.02	60.59 (+0.48)
ICLCite + USC (Chen et al., 2023)	Postprocess	10	-	57.92	37.16	62.05	60.00	54.28 (-1.18)	63.27	40.75	68.90	67.60	60.13 (+0.02)
CaLM (ours)		≤ 4	≤ 3	**77.18**	**42.24**	**63.71**	**64.99**	**62.03** (+6.57)	**82.08**	**44.21**	**70.55**	**72.37**	**67.30** (+7.19)
Finetune Llama-2 Baseline													
Self-RAG (7B) †	Finetune LM	-	-	-	-	-	-	-	74.3	30.0	66.9	67.8	59.8
Self-RAG (13B) †	Finetune LM	-	-	-	-	-	-	-	71.6	31.7	70.3	71.3	61.2

Table 2: The experimental results on ASQA. CaLM achieves an average improvement of over 6% when using text-unicorn. When using GPT-3.5-Turbo-1106 as the main LM, CaLM is the only method that outperforms the ICLCite baseline while making the fewest total LM API calls. The best results are bold, while the second best are underlined. *USC stands for Universal Self Consistency (Chen et al., 2023). †We report Self-RAG's numbers using the results from their original paper, where they retrieve up to ten documents per input using Contriever as the retriever (Izacard et al., 2022).

Method	Method Type	# Main LM Call	# Verifier LM Call	DPR as retriever: Correctness Prec.	Correctness Rec.-5	Citation Prec.	Citation Rec.	Average	GTR as retriever: Correctness Prec.	Correctness Rec.-5	Citation Prec.	Citation Rec.	Average
GPT-3.5-Turbo-1106 as Main LM													
ICLCite (Gao et al., 2023b)	Single Run	1	-	12.47	10.28	12.60	11.62	11.74	19.23	17.32	21.75	20.77	19.77
Summ + ICLCite (Gao et al., 2023b)	Preprocess	10	-	6.65	4.97	10.56	9.43	7.90 (-3.84)	15.06	13.10	21.89	21.00	17.76 (-2.01)
Snippet + ICLCite (Gao et al., 2023b)		10	-	11.67	7.87	**14.14**	12.64	11.58 (-0.16)	20.48	17.18	**25.45**	23.91	21.76 (+1.99)
ICLCite + USC (Chen et al., 2023)*	Postprocess	10	-	9.88	8.36	10.14	9.28	9.42 (-2.02)	14.07	12.45	17.34	16.60	15.12 (-4.65)
CaLM (ours)		≤ 4	≤ 3	**17.65**	**13.61**	13.93	**12.99**	**14.55** (+2.81)	**26.71**	**18.55**	25.16	**24.39**	**23.70** (+3.93)
text-unicorn as Main LM													
ICLCite (Gao et al., 2023b)	Single Run	1	-	17.06	12.55	15.77	15.59	15.24	26.27	21.42	25.52	25.21	24.61
Summ + ICLCite (Gao et al., 2023b)	Preprocess	10	-	18.61	15.17	15.61	15.36	16.19 (+0.95	27.11	**26.14**	24.61	24.98	25.71 (+1.10)
Snippet + ICLCite (Gao et al., 2023b)		10	-	18.81	15.11	**16.10**	15.83	16.46 (+1.22)	27.26	25.07	25.96	25.55	25.96 (+1.35)
ICLCite + USC (Chen et al., 2023)	Postprocess	10	-	17.04	12.47	15.81	15.62	15.24 (+0.0)	26.34	22.10	25.48	25.19	24.78 (+0.17)
CaLM (ours)		≤ 4	≤ 3	**19.62**	**16.71**	15.90	**15.95**	**17.05** (+1.81)	**30.28**	22.79	**28.53**	**28.34**	**27.49** (+2.88)

Table 3: The experimental results on QAMPARI. Compared to all the preprocess and postprocess baselines, our method obtains the best average performance across different settings and use the fewest LM API calls.

Method	Fluency mauve	Correct. Claim.	Citation Prec.	Citation Rec.	Average
GPT-3.5-Turbo-1106 as Main LM					
ICLCite	25.33	12.90	44.63	49.34	33.05
Summ + ICLCite	17.86	10.67	38.81	43.26	26.40 (-6.65)
Snippet + ICLCite	25.30	12.06	35.49	41.35	28.55 (-4.50)
ICLCite + USC	25.04	12.16	34.87	39.60	27.92 (-5.13)
CaLM (ours)	**25.84**	**12.92**	**46.55**	**51.90**	**34.30** (+1.25)
text-unicorn as Main LM					
ICLCite	35.82	12.21	35.35	32.05	28.86
Summ + ICLCite	34.57	11.74	35.54	35.02	29.22 (+0.36)
Snippet + ICLCite	**47.43**	**13.51**	34.37	33.00	32.08 (+3.22)
ICLCite + USC	31.42	12.29	34.90	31.66	27.57 (-1.29)
CaLM (ours)	37.06	11.95	**46.26**	**43.60**	**34.72** (+5.86)

Table 4: The experimental results on ELI5. CaLM obtains the best average performance regardless of the used main LM. The improvements are especially significant in the citation quality.

We observe that with a weaker retriever, Snippet+ICLCite often fails to enhance performance. We conjecture that this is attributed to increased noise in these scenarios. More noisy input lists can lead to a higher likelihood of hallucinations and errors during the preprocessing steps, resulting in degraded performance.

3. CaLM effectively improves the performance in both answer correctness and citation quality, and the improvement is robust against the choice of retriever. We attribute this robustness to our approach of releasing only high-quality responses in each iteration while continuously exploring new batches of documents.

6.2 Analysis

We conduct analysis of CaLM on the QAMPARI dataset. All the studies are conducted using GTR as the retriever and text-unicorn as the main LM, except where noted in the table.

What if we use larger LM as the verifier LM? Tab. 5 shows how the size of the verifier impacts

Choice of Verifier LM.	Correctness Prec.	Correctness Rec.-5	Citation Prec.	Citation Rec.	Average
GPT-3.5-Turbo-1106 as Main LM					
No verification	19.23	17.32	21.75	20.77	19.77
Tulu-2-dpo-7b	25.36	17.95	24.29	23.64	22.81 (+3.04)
Tulu-2-dpo-13b (our choice)	**26.71**	18.55	**25.16**	**24.39**	**23.70** (+3.93)
Tulu-2-dpo-70b	25.61	**18.69**	23.64	23.57	22.88 (+3.11)
GPT-3.5-Turbo-1106	24.39	18.36	22.94	22.30	22.00 (+2.23)
text-unicorn as Main LM					
No verification	26.27	21.42	25.52	25.21	24.61
Tulu-2-dpo-7b	29.71	21.80	27.67	27.45	26.65 (+2.04)
Tulu-2-dpo-13b (our choice)	**30.28**	22.79	**28.53**	**28.34**	**27.49** (+2.88)
Tulu-2-dpo-70b	28.94	22.61	27.06	26.88	26.37 (+1.76)
text-unicorn	29.73	**23.44**	28.01	27.54	27.18 (+2.57)

Table 5: Evaluating the choice of verifier LM. Our investigation focuses on comparing the performance of models of varying sizes and using the main LM itself as the verifier. Compared to the largest LM in the same family, a smaller-sized LM yields better performance.Additionally, our choice of verifier can largely outperform using the main LM itself as verifier.

Verifier LM's input document	Correctness Prec.	Correctness Rec.-5	Citation Prec.	Citation Rec.	Average
Only cited documents (Ours.)	30.28	22.79	28.53	28.34	27.49
Same as the main LM	28.19	23.26	25.22	25.65	25.58

Table 6: Ablating our design of using only cited documents for verification. We can observe a significant performance drop if we remove the design.

the task performance. We can see that all verifiers improve the performance, demonstrating the robustness of our framework. Additionally, the smaller LM variants outperforms the largest LM (tulu-2-70b) in citation quality, and the medium-sized LM achieves the best performance. A medium-sized LM combines external knowledge integration and information discernment, resulting in a better verifier performance. Comparing with using the large LM itself as verifier, our choice of a smaller LM as the verifier performs better.

What if the verifier LM could access more than just the cited documents? In our automated verification method, the verifier LM is limited to accessing only the cited documents. Removing this constraint simplifies our verification algorithm to re-sampling with a different LM. The results, shown in Tab. 6, demonstrate a notable decline in performance when such verification design is removed, especially in citation quality. This highlights the essential role and effectiveness of our design of automated verification.

Further Analysis Appendix A details additional studies on model performance and prediction changes across iterations.

7 Related Work

Evaluation Early research focused on evaluating attribution in text generation. Rashkin et al. (2023) introduced the "Attributable to Identified Sources" (AIS) framework for assessing faithfulness. Subsequent studies developed automatic (Honovich et al., 2022; Yue et al., 2023) and human (Bohnet et al., 2022) evaluation methods based on AIS. Recent work by Liu et al. (2023b) evaluated generative search engines that provide citations, and Gao et al. (2023b) proposed ALCE, an automatic benchmark for text generation with citations. In this study, we assess our approach using ALCE.

Finetuned LMs Some studies have investigated fine-tuning language models (LMs) for generating cited answers (Menick et al., 2022; Nakano et al., 2021). Similarly, Ye et al. (2023) employed adaptation approach for fine-tuning. These methods required training and could be be susceptible to generalization issues.

Retrieval-based Methods He et al. (2022); Gao et al. (2023a) used post-editing to ensure text consistency by retrieving relevant documents. Gao et al. (2023b) explored methods like document summarization and LLM-enabled searches for citation improvement, yet lacked verification for answer validation. Li et al. (2023) utilized an LLM as a verifier for document relevance but didn't employ the answer to verify the correct grounding.

Self-reflection Prompting LLMs to self-reflect on their answers has been shown to improve factuality (Ji et al., 2023). Asai et al. (2023) employed this concept, enhancing LM quality and factuality via retrieval and self-reflection by training special tokens. CaLM outperforms this method without the need of training.

8 Conclusion

In this paper, we introduce CaLM, a novel verification approach for grounded generation. We observe that while larger LMs excel at identifying relevant materials, they tend to rely excessively on internal parametric memory. Conversely, smaller LMs are adept at processing focused information. CaLM leverages these complementary strengths to offer a fresh perpective on robust and scalable solution for verification in grounded generation.

Limitation

We acknowledge the limitations of CaLM from the following aspects to inspire future research opportunities in the field of grounded generation.

Firstly, as a postprocessing technique, our method introduces additional latency in generating the final answer. As a remedy, we can set the maximum iteration T smaller. From Fig. 4 in the appendix, we have shown that only even a single iteration of our correction process significantly enhances performance, yet latency remains an unavoidable factor.

Moreover, unlike preprocessing approaches depicted in Fig. 1(b), which can reduce input token consumption for the final LLM, our method necessitates that the LLM initially processes all documents, leading to a higher cost for token usage.

Lastly, despite the considerable advancements CaLM has made across datasets, the instances that pass our verification process are still not flawless. Given that both the answer candidate and the verifier output are outcomes from LMs, there is an inevitable risk of both models producing hallucinations simultaneously.

We hope future works can leverage the idea and insights from CaLM to advance the development of more robust grounded generation with low latency and reduced token costs.

Broader Considerations

As a method that directly apply LLMs, CaLM inherits all potential risks associated with LLMs, including but not limited to unethical outputs, toxicity, and biases (Bender et al., 2021; Yuan et al., 2024; Gallegos et al., 2023). Our qualitative assessment of CaLM, conducted across several samples from three datasets, indicates that LLMs generally adhere to instructions and generate responses relevant to the content of provided documents. However, we strongly advise conducting a comprehensive evaluation of these potential issues before deploying CaLM in practical settings.

References

Samuel Joseph Amouyal, Ohad Rubin, Ori Yoran, Tomer Wolfson, Jonathan Herzig, and Jonathan Berant. 2022. QAMPARI: : An open-domain question answering benchmark for questions with many answers from multiple paragraphs. *arXiv preprint arXiv:2205.12665*.

Rohan Anil, Andrew M. Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, Eric Chu, Jonathan H. Clark, Laurent El Shafey, Yanping Huang, Kathy Meier-Hellstern, Gaurav Mishra, Erica Moreira, Mark Omernick, Kevin Robinson, Sebastian Ruder, Yi Tay, Kefan Xiao, Yuanzhong Xu, Yujing Zhang, Gustavo Hernández Ábrego, Junwhan Ahn, Jacob Austin, Paul Barham, Jan A. Botha, James Bradbury, Siddhartha Brahma, Kevin Brooks, Michele Catasta, Yong Cheng, Colin Cherry, Christopher A. Choquette-Choo, Aakanksha Chowdhery, Clément Crepy, Shachi Dave, Mostafa Dehghani, Sunipa Dev, Jacob Devlin, Mark Díaz, Nan Du, Ethan Dyer, Vladimir Feinberg, Fangxiaoyu Feng, Vlad Fienber, Markus Freitag, Xavier Garcia, Sebastian Gehrmann, Lucas Gonzalez, and et al. 2023. Palm 2 technical report. *CoRR*, abs/2305.10403.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2023. Self-rag: Self-reflective retrieval augmented generation. In *NeurIPS 2023 Workshop on Instruction Tuning and Instruction Following*.

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? In *FAccT '21: 2021 ACM Conference on Fairness, Accountability, and Transparency, Virtual Event / Toronto, Canada, March 3-10, 2021*.

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

Xinyun Chen, Renat Aksitov, Uri Alon, Jie Ren, Kefan Xiao, Pengcheng Yin, Sushant Prakash, Charles Sutton, Xuezhi Wang, and Denny Zhou. 2023. Universal self-consistency for large language model generation. *arXiv preprint arXiv:2311.17311*.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 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*.

Isabel O. Gallegos, Ryan A. Rossi, Joe Barrow, Md. Mehrab Tanjim, Sungchul Kim, Franck Dernoncourt, Tong Yu, Ruiyi Zhang, and Nesreen K. Ahmed. 2023. Bias and fairness in large language models: A survey. *CoRR*, abs/2309.00770.

Luyu Gao, Zhuyun Dai, Panupong Pasupat, Anthony Chen, Arun Tejasvi Chaganty, Yicheng Fan, Vincent Zhao, Ni Lao, Hongrae Lee, Da-Cheng Juan, and Kelvin Guu. 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).*

Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. 2023b. Enabling large language models to generate text with citations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*.

Hangfeng He, Hongming Zhang, and Dan Roth. 2022. Rethinking with retrieval: Faithful large language model inference. *arXiv preprint arXiv:2301.00303*.

Or Honovich, Roee Aharoni, Jonathan Herzig, Hagai Taitelbaum, Doron Kukliansy, 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*.

Jie Huang and Kevin Chen-Chuan Chang. 2023. Citation: A key to building responsible and accountable large language models. *arXiv preprint arXiv:2307.02185*.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2023. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *arXiv preprint arXiv:2311.05232*.

Hamish Ivison, Yizhong Wang, Valentina Pyatkin, Nathan Lambert, Matthew Peters, Pradeep Dasigi, Joel Jang, David Wadden, Noah A. Smith, Iz Beltagy, and Hannaneh Hajishirzi. 2023. Camels in a changing climate: Enhancing LM adaptation with tulu 2. *arXiv preprint arXiv:2311.10702*.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022. Unsupervised dense information retrieval with contrastive learning. *Trans. Mach. Learn. Res.*

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

Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. 2023. Llm-blender: Ensembling large language models with pairwise ranking and generative fusion. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick S. H. 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 2020, Online, November 16-20, 2020*.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Xiaonan Li, Changtai Zhu, Linyang Li, Zhangyue Yin, Tianxiang Sun, and Xipeng Qiu. 2023. Llatrieval: Llm-verified retrieval for verifiable generation. *arXiv preprint arXiv:2311.07838*.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023a. Lost in the middle: How language models use long contexts. ArXiv:2307.03172.

Nelson F. Liu, Tianyi Zhang, and Percy Liang. 2023b. Evaluating verifiability in generative search engines. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*.

Jacob Menick, Maja Trebacz, Vladimir Mikulik, John Aslanides, Francis Song, Martin Chadwick, Mia Glaese, Susannah Young, Lucy Campbell-Gillingham, Geoffrey Irving, et al. 2022. Teaching language models to support answers with verified quotes, 2022. *URL https://arxiv.org/abs/2203.11147*.

Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2020. Ambigqa: Answering ambiguous open-domain questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*.

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*.

Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernández Ábrego, Ji Ma, Vincent Y. Zhao, Yi Luan, Keith B. Hall, Ming-Wei Chang, and Yinfei Yang. 2022. Large dual encoders are generalizable retrievers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick S. H. Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, Vassilis Plachouras, Tim Rocktäschel, and Sebastian Riedel. 2021. KILT: a benchmark for knowledge intensive language tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*.

Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Dmytro Okhonko, Samuel Broscheit, Gautier Izacard, Patrick S. H. Lewis, Barlas Oguz, Edouard Grave, Wen-tau Yih, and Sebastian Riedel. 2021. The web is your oyster - knowledge-intensive NLP against a very large web corpus.

Krishna Pillutla, Swabha Swayamdipta, Rowan Zellers, John Thickstun, Sean Welleck, Yejin Choi, and Zaïd Harchaoui. 2021. MAUVE: measuring the gap between neural text and human text using divergence frontiers. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*.

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*.

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*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*.

Jian Xie, Kai Zhang, Jiangjie Chen, Renze Lou, and Yu Su. 2023. Adaptive chameleon or stubborn sloth: Unraveling the behavior of large language models in knowledge clashes. *arXiv preprint arXiv:2305.13300*.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. 2023. Wizardlm: Empowering large language models to follow complex instructions. *arXiv preprint arXiv:2304.12244*.

Xi Ye, Ruoxi Sun, Sercan Ö Arik, and Tomas Pfister. 2023. Effective large language model adaptation for improved grounding. *arXiv preprint arXiv:2311.09533*.

Tongxin Yuan, Zhiwei He, Lingzhong Dong, Yiming Wang, Ruijie Zhao, Tian Xia, Lizhen Xu, Binglin Zhou, Fangqi Li, Zhuosheng Zhang, Rui Wang, and Gongshen Liu. 2024. R-judge: Benchmarking safety risk awareness for LLM agents. *CoRR*, abs/2401.10019.

Xiang Yue, Boshi Wang, Ziru Chen, Kai Zhang, Yu Su, and Huan Sun. 2023. Automatic evaluation of attribution by large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*.

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, Longyue Wang, Anh Tuan Luu, Wei Bi, Freda Shi, and Shuming Shi. 2023. Siren's song in the AI ocean: A survey on hallucination in large language models. *arXiv preprint arXiv:2309.01219*.

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. *arXiv preprint arXiv:2306.05685*.

A Further Analysis

In this section, we focus on analysis when iteration of CaLM proceed.

How does the model's performance improve as iterations proceed? Fig. 4 illustrates the outcomes of terminating the process from iteration 0 through 6. As demonstrated in the figure, the precision for correctness consistently improve with each iteration of CaLM as our framework only allows high quality final output. On the other hand, our design of updating the input document list contributes to a consistent rise in correctness recall.

However, we observe that performance tends to plateau after the third iteration, with subsequent iterations yielding diminishing returns. Extending the maximum iterations to six produced only a marginal average improvement of 0.4 compared to the third iteration, while significantly increasing the computational cost in terms of API calls.

We believe CaLM's iterative nature is a key strength, allowing for continuous improvement. However, our findings suggest that two to three iterations offer substantial gains with minimal computational overhead. This demonstrates the framework's efficiency and practicality for real-world applications.

Case study on CaLM's correction. We present two case studies to demonstrate that by looping through our verification design, more accurate evidence can be found and more accurate responses can be generated. Fig. 5 is based on the ASQA dataset, and Fig. 6 utilizes the QAMPARI dataset.

From the example of Fig. 5, we can see that the small LM, serving as a verifier, when given access to cleaner input document sets, is capable of identifying overlooked information by the main LM. This detection initiates iterative correction processes in subsequent rounds.

The example in Fig. 6 demonstrates that (1) CaLM finds more convincing evidence documents in later rounds, and (2) CaLM catches citation errors through verification.

B Used prompts

In this section, we list the prompts we use for our experiment.

B.1 Prompts for ASQA

Two different prompt sets are used for the ASQA dataset. Fig. 7 shows the prompts we used for the LM to conduct grounded generation, which mainly follow the prompt used in (Gao et al., 2023b) with two shot examples. We design our own prompt for the main model to perform correction. The prompt is detailed in Fig. 8, which use 1-shot example.

B.2 Prompts for QAMPARI

Two different prompt sets are used for the QAMPRI dataset. Fig. 9 shows the prompts we used for the LM to conduct grounded generation, which mainly follow the prompt used in (Gao et al., 2023b) with two shot examples. We design our own prompt for the main model to perform correction. The prompt is detailed in Fig. 10, which use 1-shot example.

B.3 Prompts for ELI5

Two different prompt sets are used for the ELI5 dataset. Fig. 11 shows the prompts we used for the LM to conduct grounded generation, which mainly follow the prompt used in (Gao et al., 2023b) with two shot examples. We design our own prompt for the main model to perform correction. The prompt is detailed in Fig. 12, which use 1-shot example.

C Implementation Details

For all experiments with public available models, we use vLLM framework for inference (Kwon et al., 2023). We operate vLLM on our machine with 16 NVIDIA-A100-40GB GPU. For experiments with GPT-3.5-Turbo-1106, we use the official API [3]. For experiment with text-unicorn, we use Google-Cloud vertex API [4].

D Dataset and Evaluation Tool

We use the artifacts provided by Gao et al. (2023b). The dataset and corresponding evaluation code is under MIT licence [5]. We do not change any of the provided data and maintain consistent with their intended use.

[3]https://platform.openai.com/
[4]https://cloud.google.com/vertex-ai/docs/reference/rest
[5]https://github.com/princeton-nlp/ALCE/tree/main

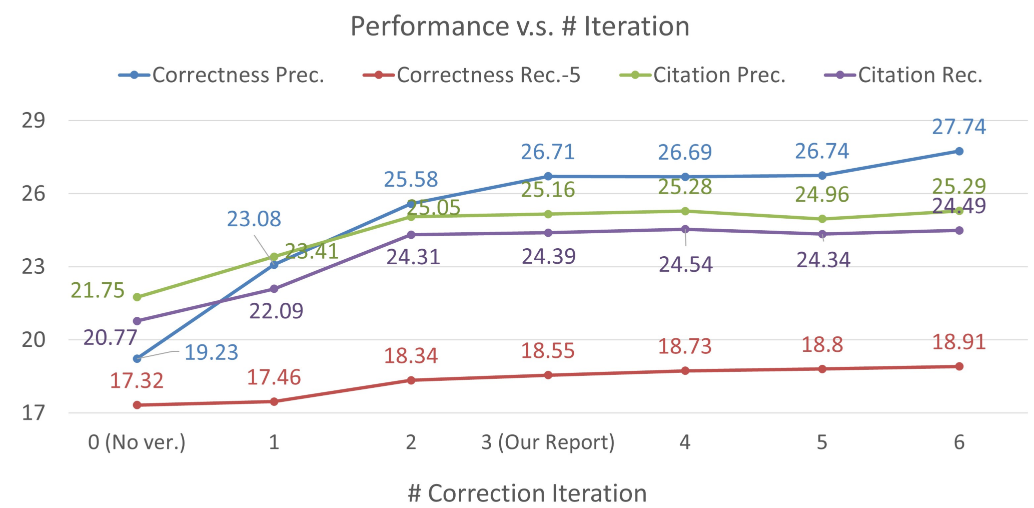

Figure 4: The study examines the iterative performance improvements on the QAMPARI dataset. We use GPT-3.5-Turbo as the main LM for this running study.

```
> Round 1:

>> Main LM:
>>> Access to:
Document [1](Title: Don't Tell Me What to Do): "Don't Tell Me What to Do" is a song written by Harlan Howard
 and Max D. Barnes, and recorded by American country music artist Pam Tillis. It was released in December
1990 as the first single from the album "Put Yourself in My Place". The song reached number 5 on the "
Billboard" Hot Country Singles & Tracks chart. Marty Stuart recorded this song under the title "I'll Love
You Forever (If I Want To)" in 1988

Document [2](Title: Don't (Ed Sheeran song)): (Omitted)

Document [3](Title: Don't Speak): (Omitted)

Document [4](Title: Don't Tell Me What to Do): for his "Let There Be Country" album, although the album was
not released until 1992. Don't Tell Me What to Do "for Baby Animals' song of the same name, see "Don't Tell
Me What to Do (Baby Animals song)". "Don't Tell Me What to Do" is a song written by Harlan Howard and Max D.
 Barnes, and recorded by American country music artist Pam Tillis. It was released in December 1990 as the
first single from the album ...(Omitted)

Document [5](Title: Pam Tillis): Pam Tillis Pamela Yvonne Tillis (born July 24, 1957) is an American country
 music singer-songwriter and actress. She is the daughter of the late country music singer Mel Tillis and
Doris Tillis. ...(Omitted)... By 1991, she had signed to Arista Nashville, entering Top 40 on Hot Country
Songs for the first time with "Don't Tell Me What to Do", the first of five

>>> Response:
The song "Don't Tell Me What to Do" was written by Harlan Howard and Max D. Barnes and recorded by American
country music artist Pam Tillis, reaching number 5 on the "Billboard" Hot Country Singles & Tracks chart
[1][4][5].

>> Verifier LM:
>>> Access to: Document 1,4,5
>>> Response:
Pam Tillis recorded the song "Don't Tell Me What to Do" [5], which reached the Top 40 of Hot Country Songs
in 1991 [5]. This song was also written by Harlan Howard and Max D. Barnes, and Marty Stuart recorded a
version of this song [1][4].

> Round 2:

>> Main LM:
>>> Access to:
Document [1](Title: Don't Tell Me What to Do): "Don't Tell Me What to Do" is a song written by Harlan Howard
 and Max D. Barnes, and recorded by American country music artist Pam Tillis. It was released in December
1990 as the first single from the album "Put Yourself in My Place". The song reached number 5 on the "
Billboard" Hot Country Singles & Tracks chart. Marty Stuart recorded this song under the title "I'll Love
You Forever (If I Want To)" in 1988

Document [4](Title: Don't Tell Me What to Do): for his "Let There Be Country" album, although the album was
not released until 1992. Don't Tell Me What to Do "for Baby Animals' song of the same name, see "Don't Tell
Me What to Do (Baby Animals song)". "Don't Tell Me What to Do" is a song written by Harlan Howard and Max D.
 Barnes, and recorded by American country music artist Pam Tillis. It was released in December 1990 as the
first single from the album ...(Omitted)

Document [5](Title: Pam Tillis): Pam Tillis Pamela Yvonne Tillis (born July 24, 1957) is an American country
 music singer-songwriter and actress. She is the daughter of the late country music singer Mel Tillis and
Doris Tillis. ...(Omitted)... By 1991, she had signed to Arista Nashville, entering Top 40 on Hot Country
Songs for the first time with "Don't Tell Me What to Do", the first of five

Document [6](Title: Don't Speak): (Omitted)

Document [7](Title: Don't (Elvis Presley song)): (Omitted)

>>> Response:
The song "Don't Tell Me What to Do" was written by Harlan Howard and Max D. Barnes and recorded by American
country music artist Pam Tillis, reaching number 5 on the "Billboard" Hot Country Singles & Tracks chart
[1][4][5]. Additionally, Marty Stuart also recorded this song under the title "I'll Love You Forever (If I
Want To)" in 1988 [1].

(Omit the rest of the steps for clarity.)
```

Figure 5: A case study of CaLM on ASQA dataset. The question is *"Who sings don't tell me what to do?"* and all reference short answers are *"Pam Tillis"*, *"Marty Stuart"*, and *"Baby Animals"*.

```
> Round 1:

>> Main LM:
>>> Access to:
Document [1](Title: James Newton Howard):  (Omitted)

Document [2](Title: John Carpenter): in October 2018. The film acts as a direct sequel to Carpenter's
original film, ignoring the continuity of all other previous films. It is his first direct involvement with
the franchise since 1982's ". Carpenter's films are characterized by minimalist lighting and photography,
static cameras, use of steadicam, and distinctive synthesized scores (usually self-composed). With the
exception of "Someone's Watching Me!", "Elvis", "The Thing", "Starman", "Memoirs of an Invisible Man", and "
The Ward", he has scored all of his films (though some are collaborations), most famously the themes from "
Halloween" and "Assault on Precinct 13". His music is generally synthesized

Document [3](Title: Thomas Newman):  (Omitted)

Document [4](Title: John Carpenter): a score by Ennio Morricone and a cast including young actor Kurt
Russell and respected character actors such as Wilford Brimley, Richard Dysart, Charles Hallahan, Keith
David, and Richard Masur. "The Thing" was distributed by Universal Pictures. Although Carpenter's film used
the same source material as the 1951 Howard Hawks film, "The Thing from Another World", it is more faithful
to the John W. Campbell, Jr. novella, "Who Goes There?", upon which both films were based. Moreover, unlike
the Hawks film, "The Thing" was part of what Carpenter later called his "Apocalypse Trilogy," a trio of
films ("The Thing", "Prince

Document [5](Title: Alan Howarth (composer)): (Omitted)

>>> Response:
Assault on Precinct 13 [2], Halloween [2], The Thing [4].

>> Verifier LM:
>>> Access to: Document 2,4
>>> Response:
Assault on Precinct 13 [2].

> Round 2:

>> Main LM:
>>> Access to:
Document [2](Title: John Carpenter): in October 2018. The film acts as a direct sequel to Carpenter's
original film, ignoring the continuity of all other previous films. It is his first direct involvement with
the franchise since 1982's ". Carpenter's films are characterized by minimalist lighting and photography,
static cameras, use of steadicam, and distinctive synthesized scores (usually self-composed). With the
exception of "Someone's Watching Me!", "Elvis", "The Thing", "Starman", "Memoirs of an Invisible Man", and "
The Ward", he has scored all of his films (though some are collaborations), most famously the themes from "
Halloween" and "Assault on Precinct 13". His music is generally synthesized

Document [4](Title: John Carpenter): a score by Ennio Morricone and a cast including young actor Kurt
Russell and respected character actors such as Wilford Brimley, Richard Dysart, Charles Hallahan, Keith
David, and Richard Masur. "The Thing" was distributed by Universal Pictures. Although Carpenter's film used
the same source material as the 1951 Howard Hawks film, "The Thing from Another World", it is more faithful
to the John W. Campbell, Jr. novella, "Who Goes There?", upon which both films were based. Moreover, unlike
the Hawks film, "The Thing" was part of what Carpenter later called his "Apocalypse Trilogy," a trio of
films ("The Thing", "Prince

Document [6](Title: John Carpenter): classics that Carpenter has directed include "Dark Star" (1974), "
Assault on Precinct 13" (1976), "The Thing" (1982), "Christine" (1983), "Big Trouble in Little China" (1986)
, "Prince of Darkness" (1987), "They Live" (1988), and "In the Mouth of Madness" (1995). He returned to the
"Halloween" franchise as both composer and executive producer for the horror sequel "Halloween" (2018).
Carpenter is also notable for having composed or co-composed most of his films' music; some of them are now
well-known, with the main theme of "Halloween" being considered a part of popular culture. He won a Saturn
Award for Best Music for the

Document [7](Title: Donald Rubinstein): (Omitted)

Document [8](Title: Nellee Hooper): (Omitted)

>>> Response:
Halloween [6], Dark Star [6], Assault on Precinct 13 [6], The Thing [6], Christine [6], Big Trouble in
Little China [6], Prince of Darkness [6], They Live [6], In the Mouth of Madness [6]

>> Verifier LM:
>>> Access to: Document 6
>>> Response:
Halloween [6], Dark Star [6], The Thing [6], Christine [6], Big Trouble in Little China [6], Prince of
Darkness [6], They Live [6], In the Mouth of Madness [6]

(Omit the rest of the steps for clarity.)
```

Figure 6: A case study of CaLM on QAMPARI dataset. The question is *"Which movie did John Carpenter direct for which he also composed the music?"* and all reference answers are *"Vampires", "In the Mouth of Madness", "Assault on Precinct 13", "Dark Star", "Big Trouble in Little China", "They Live", "Halloween", "Escape from New York", "Prince of Darkness", "Ghosts of Mars", "The Fog", "Chevil", "Village of the Damned"*.

```
Instruction: Write an accurate, engaging, and concise answer for the given question using only the provided
search results (some of which might be irrelevant) and cite them properly. Use an unbiased and journalistic
tone. Always cite for factual claims. When citing several search results, use [1][2][3]. Cite at least one
and no more than three documents per sentence. If multiple documents support the sentence, only cite a
minimum sufficient subset.

Question: {EXAMPLE 1}

Document [1](Title: {TITLE_FOR_DOC_1}): {DOC_1}
Document [2](Title: {TITLE_FOR_DOC_2}): {DOC_2}
Document [3](Title: {TITLE_FOR_DOC_3}): {DOC_3}
Document [4](Title: {TITLE_FOR_DOC_4}): {DOC_4}
Document [5](Title: {TITLE_FOR_DOC_5}): {DOC_5}


Answer:  {ANSWER_FOR_EXAMPLE 1}


Instruction: Write an accurate, engaging, and concise answer for the given question using only the provided
search results (some of which might be irrelevant) and cite them properly. Use an unbiased and journalistic
tone. Always cite for factual claims. When citing several search results, use [1][2][3]. Cite at least one
and no more than three documents per sentence. If multiple documents support the sentence, only cite a
minimum sufficient subset.

Question: {EXAMPLE 2}

Document [1](Title: {TITLE_FOR_DOC_1}): {DOC_1}
Document [2](Title: {TITLE_FOR_DOC_2}): {DOC_2}
Document [3](Title: {TITLE_FOR_DOC_3}): {DOC_3}
Document [4](Title: {TITLE_FOR_DOC_4}): {DOC_4}
Document [5](Title: {TITLE_FOR_DOC_5}): {DOC_5}


Answer:  {ANSWER_FOR_EXAMPLE 2}


Instruction: Write an accurate, engaging, and concise answer for the given question using only the provided
search results (some of which might be irrelevant) and cite them properly. Use an unbiased and journalistic
tone. Always cite for factual claims. When citing several search results, use [1][2][3]. Cite at least one
and no more than three documents per sentence. If multiple documents support the sentence, only cite a
minimum sufficient subset.

Question: {REAL QUERY}

Document [1](Title: {TITLE_FOR_DOC_1}): {DOC_1}
Document [2](Title: {TITLE_FOR_DOC_2}): {DOC_2}
Document [3](Title: {TITLE_FOR_DOC_3}): {DOC_3}
Document [4](Title: {TITLE_FOR_DOC_4}): {DOC_4}
Document [5](Title: {TITLE_FOR_DOC_5}): {DOC_5}


Answer:
```

Figure 7: Prompt for the LM to conduct the grounded generation on ASQA dataset.

```
Instruction: Provide a concise response to the question by analyzing relevant search results (some of which
might be irrelevant), and cite useful resources using [1][2][3] format. Use an unbiased and journalistic
tone, ensuring facts are presented clearly based on the search documents. Cite at least one and no more than
 three documents per sentence. Additionally, you will be provided with an incomplete draft solution, which
is based on the first {SIZE_OF_VERIFIED_DOCS} search results and might contain citation inaccuracies.
Therefore, it might not capture all conceivable responses to the question. Your role is to assess the draft'
s comprehensiveness as well as its correctness, and then update the solution to encapsulate all possible
answers according to the search documents. Provide your answer after "Corrected Answer:", and ensure each
sentence is supported by citations from one to three sources.

Question: {EXAMPLE 1}

Document [1](Title: {TITLE_FOR_DOC_1}): {DOC_1}
Document [2](Title: {TITLE_FOR_DOC_2}): {DOC_2}
Document [3](Title: {TITLE_FOR_DOC_3}): {DOC_3}
Document [4](Title: {TITLE_FOR_DOC_4}): {DOC_4}
Document [5](Title: {TITLE_FOR_DOC_5}): {DOC_5}

Drafted Solution: {DRAFT SOLUTION FOR EXAMPLE 1}

Corrected Answer:  {ANSWER_FOR_EXAMPLE 1}

Instruction: Provide a concise response to the question by analyzing relevant search results (some of which
might be irrelevant), and cite useful resources using [1][2][3] format. Use an unbiased and journalistic
tone, ensuring facts are presented clearly based on the search documents. Cite at least one and no more than
 three documents per sentence. Additionally, you will be provided with an incomplete draft solution, which
is based on the first {SIZE_OF_VERIFIED_DOCS} search results and might contain citation inaccuracies.
Therefore, it might not capture all conceivable responses to the question. Your role is to assess the draft'
s comprehensiveness as well as its correctness, and then update the solution to encapsulate all possible
answers according to the search documents. Provide your answer after "Corrected Answer:", and ensure each
sentence is supported by citations from one to three sources.

Question: {QUERY}

Document [1](Title: {TITLE_FOR_DOC_1}): {DOC_1}
Document [2](Title: {TITLE_FOR_DOC_2}): {DOC_2}
Document [3](Title: {TITLE_FOR_DOC_3}): {DOC_3}
Document [4](Title: {TITLE_FOR_DOC_4}): {DOC_4}
Document [5](Title: {TITLE_FOR_DOC_5}): {DOC_5}

Drafted Solution: {DRAFT SOLUTION FOR QUERY}

Corrected Answer:
```

Figure 8: Prompt for the main LM to conduct correction on ASQA dataset.

```
Instruction: Provide a list of accurate answers for the given question using only the provided search
results (some of which might be irrelevant) and cite them properly using [1][2][3]. Always cite one and only
 one document for each answer. Separate answers by commas. For questions that have more than 5 answers,
write at least 5 answers.

Question: {EXAMPLE 1}

Document [1](Title: {TITLE_FOR_DOC_1}): {DOC_1}
Document [2](Title: {TITLE_FOR_DOC_2}): {DOC_2}
Document [3](Title: {TITLE_FOR_DOC_3}): {DOC_3}
Document [4](Title: {TITLE_FOR_DOC_4}): {DOC_4}
Document [5](Title: {TITLE_FOR_DOC_5}): {DOC_5}


Answer:  {ANSWER_FOR_EXAMPLE 1}


Instruction: Provide a list of accurate answers for the given question using only the provided search
results (some of which might be irrelevant) and cite them properly using [1][2][3]. Always cite one and only
 one document for each answer. Separate answers by commas. For questions that have more than 5 answers,
write at least 5 answers.

Question: {EXAMPLE 2}

Document [1](Title: {TITLE_FOR_DOC_1}): {DOC_1}
Document [2](Title: {TITLE_FOR_DOC_2}): {DOC_2}
Document [3](Title: {TITLE_FOR_DOC_3}): {DOC_3}
Document [4](Title: {TITLE_FOR_DOC_4}): {DOC_4}
Document [5](Title: {TITLE_FOR_DOC_5}): {DOC_5}


Answer:  {ANSWER_FOR_EXAMPLE 2}


Instruction: Provide a list of accurate answers for the given question using only the provided search
results (some of which might be irrelevant) and cite them properly using [1][2][3]. Always cite one and only
 one document for each answer. Separate answers by commas. For questions that have more than 5 answers,
write at least 5 answers.

Question: {REAL QUERY}

Document [1](Title: {TITLE_FOR_DOC_1}): {DOC_1}
Document [2](Title: {TITLE_FOR_DOC_2}): {DOC_2}
Document [3](Title: {TITLE_FOR_DOC_3}): {DOC_3}
Document [4](Title: {TITLE_FOR_DOC_4}): {DOC_4}
Document [5](Title: {TITLE_FOR_DOC_5}): {DOC_5}


Answer:
```

Figure 9: Prompt for the LM to conduct the grounded generation on QAMPARI dataset.

```
Instruction: Provide a list of accurate answers for the given question using only the provided search
results (some of which might be irrelevant) and cite them properly using [1][2][3]. Always cite one and only
 one document for each answer. Separate answers by commas. For questions that have more than 5 answers,
write at least 5 answers. Additionally, you will receive an incorrect draft or incomplete draft solution,
which is based on the first {SIZE_OF_VERIFIED_DOCS} search results. Hence, it may not encompass all
potential answers. Your task is to check this solution and modify it to comprehensively answer the question,
 using information from the additional provided search results. If the draft is missing, start from scratch,
 basing your answer on the search outcomes. Please begin by concisely explain your evaluation of the draft
solution within 150 words. Then, present your conclusive response under the heading "Final Answer:"

Question: {EXAMPLE 1}

Document [1](Title: {TITLE_FOR_DOC_1}): {DOC_1}
Document [2](Title: {TITLE_FOR_DOC_2}): {DOC_2}
Document [3](Title: {TITLE_FOR_DOC_3}): {DOC_3}
Document [4](Title: {TITLE_FOR_DOC_4}): {DOC_4}
Document [5](Title: {TITLE_FOR_DOC_5}): {DOC_5}

Drafted Solution: {DRAFT SOLUTION FOR EXAMPLE 1}

Corrected Answer:  {ANSWER_FOR_EXAMPLE 1}

Instruction: Provide a list of accurate answers for the given question using only the provided search
results (some of which might be irrelevant) and cite them properly using [1][2][3]. Always cite one and only
 one document for each answer. Separate answers by commas. For questions that have more than 5 answers,
write at least 5 answers. Additionally, you will receive an incorrect draft or incomplete draft solution,
which is based on the first {SIZE_OF_VERIFIED_DOCS} search results. Hence, it may not encompass all
potential answers. Your task is to check this solution and modify it to comprehensively answer the question,
 using information from the additional provided search results. If the draft is missing, start from scratch,
 basing your answer on the search outcomes. Please begin by concisely explain your evaluation of the draft
solution within 150 words. Then, present your conclusive response under the heading "Final Answer:"

Question: {QUERY}

Document [1](Title: {TITLE_FOR_DOC_1}): {DOC_1}
Document [2](Title: {TITLE_FOR_DOC_2}): {DOC_2}
Document [3](Title: {TITLE_FOR_DOC_3}): {DOC_3}
Document [4](Title: {TITLE_FOR_DOC_4}): {DOC_4}
Document [5](Title: {TITLE_FOR_DOC_5}): {DOC_5}

Drafted Solution: {DRAFT SOLUTION FOR QUERY}

Corrected Answer:
```

Figure 10: Prompt for the main LM to conduct correction on QAMPARI dataset.

```
Instruction: Write an accurate, engaging, and concise answer for the given question using only the provided
search results (some of which might be irrelevant) and cite them properly. Use an unbiased and journalistic
tone. Always cite for any factual claim. When citing several search results, use [1][2][3]. Cite at least
one document and at most three documents in each sentence. If multiple documents support the sentence, only
cite a minimum sufficient subset of the documents.

Question: {EXAMPLE 1}

Document [1](Title: {TITLE_FOR_DOC_1}): {DOC_1}
Document [2](Title: {TITLE_FOR_DOC_2}): {DOC_2}
Document [3](Title: {TITLE_FOR_DOC_3}): {DOC_3}
Document [4](Title: {TITLE_FOR_DOC_4}): {DOC_4}
Document [5](Title: {TITLE_FOR_DOC_5}): {DOC_5}


Answer:  {ANSWER_FOR_EXAMPLE 1}


Instruction: Write an accurate, engaging, and concise answer for the given question using only the provided
search results (some of which might be irrelevant) and cite them properly. Use an unbiased and journalistic
tone. Always cite for any factual claim. When citing several search results, use [1][2][3]. Cite at least
one document and at most three documents in each sentence. If multiple documents support the sentence, only
cite a minimum sufficient subset of the documents.

Question: {EXAMPLE 2}

Document [1](Title: {TITLE_FOR_DOC_1}): {DOC_1}
Document [2](Title: {TITLE_FOR_DOC_2}): {DOC_2}
Document [3](Title: {TITLE_FOR_DOC_3}): {DOC_3}
Document [4](Title: {TITLE_FOR_DOC_4}): {DOC_4}
Document [5](Title: {TITLE_FOR_DOC_5}): {DOC_5}


Answer:  {ANSWER_FOR_EXAMPLE 2}


Instruction: Write an accurate, engaging, and concise answer for the given question using only the provided
search results (some of which might be irrelevant) and cite them properly. Use an unbiased and journalistic
tone. Always cite for any factual claim. When citing several search results, use [1][2][3]. Cite at least
one document and at most three documents in each sentence. If multiple documents support the sentence, only
cite a minimum sufficient subset of the documents.

Question: {REAL QUERY}

Document [1](Title: {TITLE_FOR_DOC_1}): {DOC_1}
Document [2](Title: {TITLE_FOR_DOC_2}): {DOC_2}
Document [3](Title: {TITLE_FOR_DOC_3}): {DOC_3}
Document [4](Title: {TITLE_FOR_DOC_4}): {DOC_4}
Document [5](Title: {TITLE_FOR_DOC_5}): {DOC_5}


Answer:
```

Figure 11: Prompt for the LM to conduct the grounded generation on ELI5 dataset.

```
Instruction: Provide a succint response to the question by analyzing relevant search results (some of which
might be irrelevant), and cite these using [1][2][3] format. Limit citations to one to three per sentence,
and only cite a necessary subset if multiple sources corroborate a point. Additionally, you will be provided
 with an incomplete draft solution, which is based on the first {SIZE_OF_VERIFIED_DOCS} search results and
might contain citation inaccuracies. Therefore, it might not capture all conceivable responses to the
question. Your role is to assess the draft's comprehensiveness as well as its correctness of citations, and
then update the solution to encapsulate all possible answers according to the search documents. Provide your
 answer after "Corrected Answer:", and ensure each sentence is supported by citations from one to three
sources.

Question: {EXAMPLE 1}

Document [1](Title: {TITLE_FOR_DOC_1}): {DOC_1}
Document [2](Title: {TITLE_FOR_DOC_2}): {DOC_2}
Document [3](Title: {TITLE_FOR_DOC_3}): {DOC_3}
Document [4](Title: {TITLE_FOR_DOC_4}): {DOC_4}
Document [5](Title: {TITLE_FOR_DOC_5}): {DOC_5}

Drafted Solution: {DRAFT SOLUTION FOR EXAMPLE 1}

Corrected Answer:  {ANSWER_FOR_EXAMPLE 1}

Instruction: Provide a succint response to the question by analyzing relevant search results (some of which
might be irrelevant), and cite these using [1][2][3] format. Limit citations to one to three per sentence,
and only cite a necessary subset if multiple sources corroborate a point. Additionally, you will be provided
 with an incomplete draft solution, which is based on the first {SIZE_OF_VERIFIED_DOCS} search results and
might contain citation inaccuracies. Therefore, it might not capture all conceivable responses to the
question. Your role is to assess the draft's comprehensiveness as well as its correctness of citations, and
then update the solution to encapsulate all possible answers according to the search documents. Provide your
 answer after "Corrected Answer:", and ensure each sentence is supported by citations from one to three
sources.

Question: {QUERY}

Document [1](Title: {TITLE_FOR_DOC_1}): {DOC_1}
Document [2](Title: {TITLE_FOR_DOC_2}): {DOC_2}
Document [3](Title: {TITLE_FOR_DOC_3}): {DOC_3}
Document [4](Title: {TITLE_FOR_DOC_4}): {DOC_4}
Document [5](Title: {TITLE_FOR_DOC_5}): {DOC_5}

Drafted Solution: {DRAFT SOLUTION FOR QUERY}

Corrected Answer:
```

Figure 12: Prompt for the main LM to conduct correction on ELI5 dataset.