Ground Every Sentence: Improving Retrieval-Augmented LLMs with Interleaved Reference-Claim Generation

Anonymous EMNLP submission

Abstract

Retrieval-Augmented Generation (RAG) has been widely adopted to enhance Large Language Models (LLMs) in knowledge-intensive tasks. Recently, Attributed Text Generation (ATG) has attracted growing attention, which provides citations to support the model's responses in RAG, so as to enhance the credibility of LLM-generated content and facilitate verification. Prior methods mainly adopt coarse-grained attributions, linking to passage-level references or providing paragraph-level citations. However, these methods still fall short in verifiability and require certain time costs for fact checking. This paper proposes a fine-grained ATG method called ReClaim (**Re**fer & **Claim**), which alternates the generation of references and answers step by step. Unlike traditional coarse-grained attribution, ReClaim allows the model to add sentence-level fine-grained citations to each answer sentence in long-form question-answering tasks. Our experiments encompass various training and inference methods and multiple LLMs, verifying the effectiveness of our approach.

1 Introduction

Retrieval-Augmented Generation (RAG) (Lewis et al., 2020) is a technique that integrates information retrieval with natural language generation to enhance the performance and accuracy of large language model (LLMs) responses. However, the RAG system still encounters challenges related to verifiability and credibility. To address these issues, attributed text generation (ATG) (Bohnet et al., 2022) has been proposed. ATG aims to improve RAG systems in terms of: 1) Credibility, as explicit citations can reduce hallucinations; 2) Verifiability, making it easier for users to verify the answer.

Previous efforts on ATG mainly focus on passage-level (Thoppilan et al., 2022) or paragraph-level references (Menick et al., 2022; Nakano et al., 2021; Gao et al., 2023b). Although these attribution methods have contributed to improving the verifiability and credibility of model responses, current methods often focus on relatively coarse-grained attributions, which may contain a significant amount of irrelevant information. This increases the time cost for fact-checking.

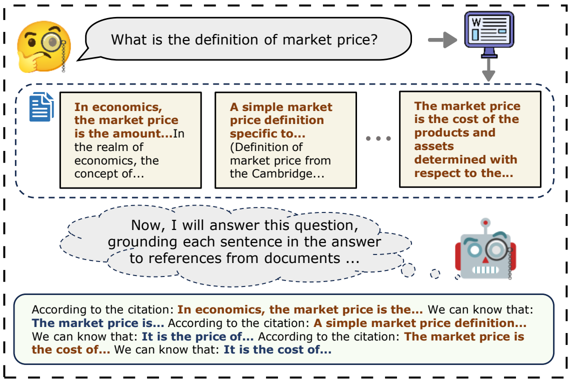
What is the definition of market price?
In economics, the market price is the amount...In the realm of economics, the concept of...
A simple market price definition specific to... (Definition of market price from the Cambridge...
...
The market price is the cost of the products and assets determined with respect to the...
Now, I will answer this question, grounding each sentence in the answer to references from documents ...
According to the citation: In economics, the market price is the... We can know that: The market price is... According to the citation: A simple market price definition... We can know that: It is the price of... According to the citation: The market price is the cost of... We can know that: It is the cost of...

Figure 1: The task setup for ReClaim. Given a question and reference passages from a large retrieval corpus. The model then generates a text response with fine-grained citations.

In this paper, we propose ReClaim, which generates attributed text with interleaving references and answers for RAG systems, as is shown in Figure 1. This method enables sentence-level fine-grained attributions for the model's answer in long-form question-answering using the RAG system.

To improve the model's text generation with citations, we constructed a training dataset, fine-tuned the model, and employed strategies for alternating citation and answer generation. We also added decoding constraints to prevent inconsistencies between citations and source reference passages.

Results of experimental results indicate that our method not only matches but also surpasses existing baseline in some metrics, particularly improving the attribution quality, which refers to the extent to which citations support the answer text. Additionally, our method reduces the length of citations, thereby reducing the effort needed for fact-checking.

Our contributions are summarized as follows:

1. We propose a method called ReClaim, which alternately generates citations and answer sentences, to enable large models to generate answer with citations.

2. For ReClaim, we constructed a training dataset and fine-tuned the model using different approaches to improve its attribution capability.

3. Through multiple experiments, we demonstrated the effectiveness of our method in enhancing the model's verifiability and credibility.

2 Related Work

Retrieval-Augmented Generation In this paper, we use the RAG (Retrieval-Augmented Generation) system to generate answer with citations. The RAG system was proposed to combine information retrieval with generation models for tasks such as question answering and knowledge generation. Similarly, this system has been widely applied to handle complex tasks that require extracting information from a large number of documents, including open-domain question answering, dialogue systems, and information summarization (Izacard and Grave, 2021; Karpukhin et al., 2020).

Long-form Text Question Answering Our work primarily focuses on the long-form question answering (LFQA) task within the RAG system. Unlike short-form QA (Rajpurkar et al., 2016; Joshi et al., 2017), which concentrates on extracting concise facts, LFQA aims to generate comprehensive answers that require a deep understanding of the context and the integration of information from multiple sources. A notable work in this field is the ELI5 (Explain Like I'm Five) dataset (Fan et al., 2019), which challenges models to provide straightforward and comprehensible explanations for complex questions. Similarly, the ASQA (Answer Summaries for Questions which are Ambiguous) dataset (Stelmakh et al., 2022) requires models to generate abstractive summaries from multiple answer passages, ensuring that the synthesized answer is both coherent and informative.

Generate Text with Citation Many current works have proposed various methods to generate answer text with citations. These methods differ in their approaches to attribution and the granularity of the citations.

LaMDA (Thoppilan et al., 2022) provides attribution for the entire response in the form of URLs pointing to entire documents. WebGPT (Nakano et al., 2021) and GopherCite (Menick et al., 2022) use reinforcement learning from human preferences to enable LLMs to answer questions while providing snippet citations. ALCE (Gao et al., 2023b) goes further by providing one or more input documents as attribution for each sentence in the answer, in a manner similar to cross-referencing.

In addition to the aforementioned methods that add citations directly during answer generation, there are some methods that focus on finding citations afterward. RARR (Gao et al., 2023a) uses two steps, Research and Revision, to add attribution to any text post hoc and to edit the original answer based on the found citations.

3 Method

This study aims to generate text with fine-grained citations by integrating citations with answers in the form of Chain-of-Thought (CoT) (Wei et al., 2022). We introduce our approach ReClaim in Section 3.1, and the following sections detail the specific implementation of ReClaim.

3.1 ReClaim: Interleaving Reference and Claim

Our task can be formally expressed as follows: Given a query q and several reference passages retrieved by the RAG system $\mathcal{D}$, the LLMs is required to generate an output $\mathcal{O}$ = { r_1, c_1, r_2, c_2...r_n, c_n }, where $\mathcal{O}$ consists of n sentence-level fine-grained references r_1, ..., r_n, which represent the fine-grained citations coming from reference passages and n claims c_1, ..., c_n, which are portions of the model's response generated based on these references. Each reference r_i corresponds to and serves as substantiation for the corresponding claim c_i. All the claims together form the model's complete answer to the question.

In our work, references are surrounded and denoted by the tags <*reference*> and </*reference*>, while claims are demarcated using the tags <*claim*> and </*claim*>. Then, they are connected in the form of CoT to produce attributed text response.

During generation, the model needs to alternately generate the reference and claim parts. In the experimental process, it was found that the model encounters several issues when generating references and claims: 1) The references generated by the model are not always completely consistent with the context of the retrieved reference passages; 2) The claims generated by the model do not always attribute well to the corresponding references. Therefore, in the following sections, we will study how to improve the generation of references and claims.

3.2 Training Dataset Construction

To improve the model's ability to generate the references and corresponding claims, we constructed a specialized fine-tuning dataset. This dataset was built based on the WebGLM-QA (Liu et al., 2023) dataset which consists of 43579 high-quality data samples for the train split. These data samples contain rich reference passages and detailed long-form answer to the question. We performed dataset construction steps and data cleansing processes on this dataset, ultimately resulting in 9433 training samples suitable for our task format.

We adopted a two-stage method to construct our training dataset: 1) Use ChatGPT (gpt-3.5-turbo) (OpenAI, 2022) to automatically segment the answers in the WebGLM-QA dataset, and identified the corresponding citations in the reference passages based on the segmented clauses; 2) Constructed the attributed answers using the clauses and their corresponding citations in the form of CoT as described in Section 3.1.

To ensure the quality of attribution in the constructed answers, we refined the preliminary training dataset: 1) Segmented the corresponding citation of each clause and performed string matching in the reference passages to filter out mismatched data items where the citation did not align with the original reference passages; 2) Used a natural language inference (NLI) (Honovich et al., 2022) model to judge the entailment relationship between the citations (as the premise) and the clauses (as the hypothesis). If there is no implication relationship, it indicates that the citation does not contain sufficient information to support the corresponding clause. Therefore, we need to filter out such data items to ensure a relatively high level of attribution quality.

Through these steps, we ultimately constructed a fine-tuned dataset that ensures text consistency of the references and high attribution quality, providing a foundation for subsequent research.

Samples	**Average Length**		
	Answer	**Citation**	**Passages**
9433	93.6	141.8	282.8

Table 1: Statistics of the training dataset.

3.3 Unified Generation

The ReClaim$_{\texttt{Unified}}$ method uses a simple fine-tuning and inference approach. It first performs instruction fine-tuning on the large language model using the dataset constructed in Section 3.2. Then, it uses the fine-tuned model to perform one-step generation. Based on the given query and reference passages, it directly outputs the attributed answer. This generation process can be described as: $UnifiedGen = \{\{r_1, c_1, ..., r_i, c_i\} \mid Query, Passages\}$.

3.4 Interleaving Generation

During the claim generation stage, since the model has already selected sufficiently granular reference text to follow, which contain the answer information, the full input context is not required. Therefore, the ReClaim w/IG (IG stands for Interleaving Generation) method trains separate models for the generation of the reference parts and the claim parts, and alternates between the two models during answer generation, adjusting the input to each model accordingly. The whole generation process is shown in Figure 2. Specifically, we adopted the following steps to train the models and generate the answer:

Reference Generation During the generation of the reference parts in the output, the model needs to generate the next reference based on the complete input context and previous output. We define this generation process as $ReferGen = \{r_{i+1} \mid Prompt, \{r_1, c_1, ..., r_i, c_i\}\}$, where r_{i+1} refers to the reference part generated in the next stage, $Prompt$ refers to the complete input context containing instructions, query and reference passages, and $r_1, c_1, ..., r_i, c_i$ refer to the previously generated references and claims. As the training of the model for generating the reference parts does

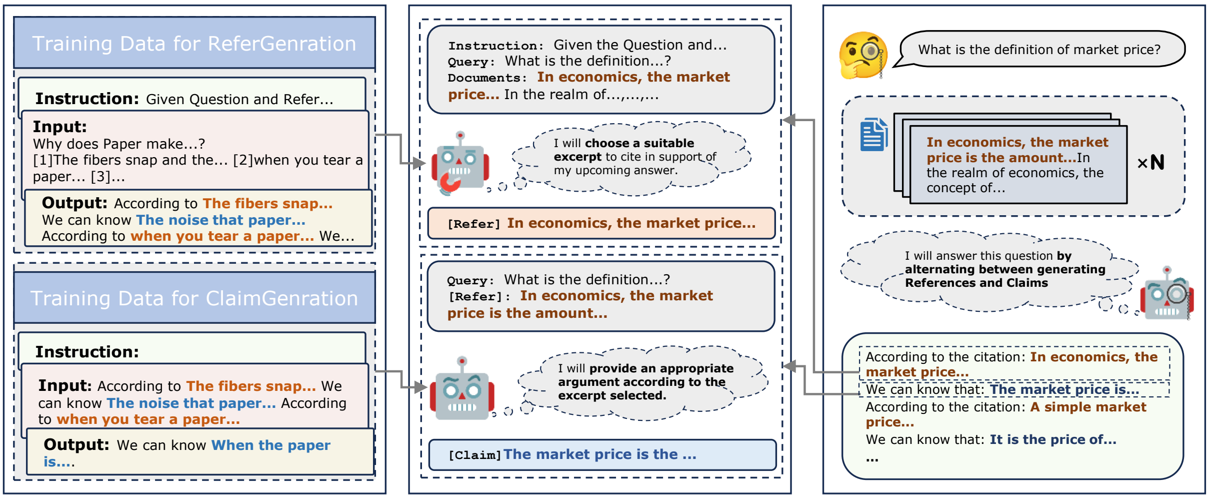
Training Data for ReferGenration
Instruction: Given Question and Refer…
Input:
Why does Paper make…?
[1]The fibers snap and the… [2]when you tear a paper… [3]…
Output: According to The fibers snap…
We can know The noise that paper…
According to when you tear a paper… We…
Training Data for ClaimGenration
Instruction:
Input: According to The fibers snap… We can know The noise that paper… According to when you tear a paper…
Output: We can know When the paper is….
Instruction: Given the Question and…
Query: What is the definition…?
Documents: In economics, the market price… In the realm of…,…,…
I will choose a suitable excerpt to cite in support of my upcoming answer.
[Refer] In economics, the market price…
Query: What is the definition…?
[Refer]: In economics, the market price is the amount…
I will provide an appropriate argument according to the excerpt selected.
[Claim] The market price is the …
What is the definition of market price?
In economics, the market price is the amount…In the realm of economics, the concept of…
×N
I will answer this question by alternating between generating References and Claims
According to the citation: In economics, the market price…
We can know that: The market price is…
According to the citation: A simple market price…
We can know that: It is the price of…
…

Figure 2: The generation process of RECLAIM w/IG. Based on the given questions and the reference passages retrieved, the model alternately generates the reference parts and the claim parts in a step-by-step manner. For these two stages of generation, distinct datasets are constructed to train the models, which alternately switches between the models and the input context during inference.

not require masking parts of the input context information, we follow the same approach as in the Section 3.3, using the training data constructed in Section 3.2 to fine-tune the model.

Claim Generation During the generation of the claim parts in the output, the model only needs to generate the next claim based on the previous output. We define this generation process as $ClaimGen = \{c_{i+1} \mid \{r_1, c_1, ..., r_i, c_i, r_{i+1}\}\}$, where c_{i+1} refers to the claim part generated in the next stage, and $r_1, c_1, ..., r_i, c_i, r_{i+1}$ refer to the previously generated references and claims. We use the training data constructed in Section 3.2 and split it into the data format that conforms to our described generation process ClaimGen. This formatted dataset is then used to fine-tune the model.

After generating the two models mentioned above, we can alternate between these models during the inference stage while simultaneously adjusting the model input according to the defined generation process. This allows us to separately generate the reference and claim parts, ultimately producing attributed text output.

3.5 Decoding Constraints

To ensure the generated reference parts align with the source reference passages, we impose decoding constraints through the following three steps:

Sentence Segmentation and Encoding We segment the reference passages into individual sentences. Then, we use the model tokenizer to encode these sentences into vectors. Each vector representation of a sentence serves as the smallest unit for generating the reference parts.

Constructing Prefix Tree The encoded vectors are transformed into a list format and organized into a Prefix tree (Fredkin, 1960) structure. Employing such a structure to store our reference sentences facilitates the choice of the next token in subsequent generation steps.

Constrained Inference During the model inference stage for generating reference parts, we select the token with the highest generation probability that satisfies the current prefix tree path as the next output token. This process continues until reaching a leaf node. Upon reaching a leaf node, the model either select another prefix tree path for output or begin the claim generation. This approach allows us to select a complete and consistent sentence from the reference passages as part of our current reference each time.

4 Experimental Protocol

In this section, we employ the GPT models and several open-source models to validate the effectiveness of our method across multiple evaluation dimensions. We conduct a comprehensive analysis by assessing the performance of our approach on various metrics.

4.1 Evaluation Datasets

Our method primarily targets long-form question answering task within the RAG system, aiming to generate attributed answer. Therefore, we selected ASQA (Stelmakh et al., 2022) and ELI5 (Fan et al., 2019) as our main benchmark datasets. By testing on these datasets, we can comprehensively assess the effectiveness of our method in generating accurate, coherent, and well-attributed long-form answers.

4.2 Evaluation Metrics

Building on our previous task definition, we focus on evaluating the model-generated outputs in three key areas:

Answer Quality We first evaluate the quality of the answers generated by the model. We concatenate all claim parts in order to form the answer to the question, and follow the ALCE evaluation method to calculate the correctness and fluency of the answers. For answer correctness, in the ASQA dataset, we use Exact Match Recall (EM Rec.) to measure the percentage of golden short answers contained in the generated answers. In the ELI5 dataset, we adopt Claim Recall (Claim Rec.) to measure the percentage of key claims included in the answers. To evaluate answer fluency, we use MAUVE (Pillutla et al., 2021) to measures the similarity between output and gold answer.

Citation Quality Similar to ALCE, we employ the AutoAIS (Bohnet et al., 2022) to measure the citation quality. Our citation quality is also measured by two metrics: 1) Correct Attribution Score (CAS), which determines if the answers is entirely supported by cited sentences; 2) Citation Redundancy Score (CRS), which identifies any redundant citation sentences.

For CAS, We use the TRUE (Honovich et al., 2022) model to compute the entailment relationship between each reference part and the corresponding claim part. The final CAS score is the proportion of sentences predicted as correctly attributed among all the sentences in the answer. For CRS, since our method allows the model to select multiple contextual reference sentences for the same claim, we need to determine if the reference contains redundant sentences. The final CRS score is the proportion of non-redundant sentences relative to all sentences in the references.

Verifiability and Credibility We employ two metrics to measure the Verifiability: 1) Citation Length, where shorter citation text typically reduces the time needed for fact-checking; 2) Attribution Ratio (AR), which represents the proportion of sentences in the output that are attributed.

We use Consistency Ratio (CR) to measure the Credibility, which determines the text consistency between the reference parts and the reference passages through string matching. Higher CR usually means the model generates fewer hallucinations when outputting the reference parts.

4.3 Methods and Baseline

To evaluate the effectiveness of the proposed method, we tested several models using various generation approaches. For each experimental setup, we used the ChatGPT results from the ALCE method as our baseline. Additionally, for the two test datasets, we adopted the Oracle-5 paragraphs provided by the ALCE as the reference passages in our model input.

Prompting We directly prompt the model to generate answer text with citations. For GPT-4o (OpenAI, 2023), we employ both zero-shot and three-shot prompting (Brown et al., 2020), while for GPT-3.5-turbo and open-source models such as Llama3-8B-Instruct (Touvron et al., 2023) and the vicuna (Chiang et al., 2023) series, we use 3-shot prompting. This approach allows us to comprehensively assess the performance and effectiveness of various models under different prompting scenarios.

Fine-tuned Model We followed the methodology described in Section 3.3 and used the comprehensive dataset constructed in Section 3.2 to conduct full fine-tuning on open-source models such as Llama3-8B-Instruct and the vicuna series models. Based on these fine-tuned models, we performed one-step text generation with citations to test the effectiveness of the ReClaim$_{\texttt{Unified}}$ method.

As outlined in Section 3.4, we fine-tune the same model, generating two separate models: ReferModel and ClaimModel. We completed our training on four 80GB A800 GPUs. For the ReferModel, we performed full fine-tuning with a learning rate of 2e-5 over 3 epochs. For the ClaimModel, we employed Lora tuning (Hu et al., 2021) with a learning rate of 5e-5 over 5 epochs. During the inference phase, we alternated between these two models to interleavingly generate the reference and claim

Method	Model	ASQA				ELI5				Average			
		Fluency	**Correct.**	**Citation**		**Fluency**	**Correct.**	**Citation**		**Fluency**	**Correct.**	**Citation**	
		MAUVE	**EM Rec.**	**CAS**	**CRS**	**MAUVE**	**Claim Rec.**	**CAS**	**CRS**	**MAUVE**	**Claim Rec.**	**CAS**	**CRS**
ALCE	VANILLA	66.8	40.4	73.6	72.5	57.2	12.0	51.1	50.0	62.0	26.2	62.4	61.3
	SUMM	70.0	43.3	68.8	61.8	40.2	12.5	51.5	48.2	55.1	27.9	60.2	55.0
	SNIPPET	69.8	41.4	65.3	57.4	62.9	14.3	50.4	45.0	66.4	27.9	57.9	51.2
	ORACLE	64.4	48.9	74.5	72.7	59.4	**21.3**	57.8	56.0	61.9	35.1	66.2	64.4
0-shot	GPT-4o	72.9	52.8	74.8	51.6	37.3	19.9	63.5	30.7	55.1	36.4	69.2	41.2
3-shot	Vicuna-7B	81.4	50.5	74.0	70.0	38.6	13.9	73.2	59.7	60.0	32.2	73.6	64.9
	Vicuna-13B	85.0	48.5	76.1	68.2	35.5	15.6	74.3	57.4	60.3	32.1	75.2	62.8
	Llama3-8B	90.1	50.7	77.7	62.1	61.3	17.9	78.3	45.3	75.7	34.3	78.0	53.7
	ChatGPT	74.9	52.6	72.5	63.4	27.8	17.8	68.6	50.8	51.4	35.2	70.6	57.1
	GPT-4o	**91.3**	**56.6**	77.4	58.0	29.7	21.1	70.2	36.8	60.5	**38.9**	73.8	47.4
ReClaim$_{\texttt{Unified}}$	Vicuna-7B	82.7	47.5	59.4	50.5	65.2	17.7	53.3	36.5	74.0	32.6	56.4	43.5
	Vicuna-13B	75.9	46.3	57.7	51.1	64.1	19.0	52.5	37.6	70.0	32.7	55.1	44.4
	Llama3-8B	88.6	50.4	65.4	57.4	**73.7**	19.4	59.3	43.8	**81.2**	34.9	62.4	50.6
ReClaim w/IG	Vicuna-13B	88.7	49.1	84.4	74.7	66.8	16.9	77.4	54.6	77.8	33.0	80.9	64.7
	Llama3-8B	86.4	50.5	**87.7**	**77.1**	66.8	16.8	**84.3**	**61.8**	76.6	33.7	**86.0**	**69.5**

Table 2: Results on ALCE benchmark (Gao et al., 2023b) and ReClaim. Definitions for Fluency, Correct. and Citation are in Section 4.2.

parts. To prevent the model from generating overly short or long answers, we set constraints on the number of reference-claim pairs. For the ASQA dataset, the model must generate between 2 to 5 pairs. For the ELI5 dataset, due to its complexity, the model must generate between 4 to 6 pairs.

To investigate the roles of the two fine-tuned models, ReferModel and ClaimModel, in the ReClaim w/IG method, we conducted ablation experiments using the Llama3-8B-Instruct model. In these experiments, we alternately used only the fine-tuned ReferModel or ClaimModel, while employing the base model with 3-shot prompting to generate the other model's output.

For these two fine-tuning methods, we adopted the constrained decoding approach described in Section 3.5 to limit the model's output in the reference parts.

5 Experiment Results

In the experiments, we wish to answer two research questions: $RQ1$) How to improve the quality of answers and citations? $RQ2$) Can ReClaim enhance the verifiability and credibility of RAG-based question answering?

5.1 How to Improve the Quality of Answers and Citations?

The overall performance is presented in Table 2:

Prompting Works Experimental results show that direct prompt models can achieve satisfactory outcomes. Compared to the ALCE method using Oracle-5 reference passages, our approach often matches or improves answer flunecy, correctness and citation correctness (CAS).

In all our prompting experiments, GPT-4o achieved the highest average correctness score (38.9) under 3-shot prompting. Additionally, Llama3-8B-Instruct achieved satisfactory results. Although it did not surpass the GPT model in terms of average correctness score, it demonstrated the best performance in fluency and CAS. Compared to the ALCE method, our model maintained response correctness while improving average fluency and CAS by 22.3% and 17.8%, respectively.

Although our method performs worse in CRS, the finer granularity of our citations minimizes the impact of redundant content. In contrast to paragraph-level citations, where each redundant citation introduces an entire paragraph of irrelevant content, our redundant citations only introduce a single sentence of irrelevant content. This does not significantly increase the fact-checking cost. Moreover, in many cases, redundant but continuous text helps in locating the citation in the original reference passages and aids in its understanding.

ReClaim$_{\texttt{Unified}}$ Cannot Improve ATG Experimental results show that while the Re-

Method	ASQA				ELI5				Average				Length
	Fluency	Correct.	Citation		Fluency	Correct.	Citation		Fluency	Correct.	Citation		
	MAUVE	EM Rec.	CAS	CRS	MAUVE	Claim Rec.	CAS	CRS	MAUVE	Claim Rec.	CAS	CRS	
ALCE	64.4	48.9	74.5	72.7	59.4	**21.3**	57.8	56.0	61.9	35.1	66.2	64.4	-
Citation-only	83.5	53.8	65.1	54.6	47.4	17.4	75.1	52.4	65.5	35.6	70.1	53.5	78.9
Claim-only	43.6	**57.7**	87.8	72.0	52.7	17.8	81.4	52.2	48.2	**37.8**	84.6	62.1	121.7
ReClaim w/IG	**86.4**	50.5	**87.7**	**77.1**	**66.8**	16.8	**84.3**	**61.8**	**76.6**	33.7	**86.0**	**69.5**	86.8

Table 3: The impact of different components under the ReClaim w/IG method on model generation results. Citation-only indicates using the fine-tuned ReferModel to generate references and the base model to generate claims. Conversely, Claim-only indicates using the fine-tuned ClaimModel to generate claims and the base model to generate references.

Method	ASQA				ELI5			
	Length		Consistency	Attri.	Length		Consistency	Attri.
	Citation	Claim	CR	AR	Citation	Claim	CR	AR
ALCE	536.3	85.5	100.0	91.3	660.0	98.09	100.0	96.9
3-shot$_{\text{UD}}$	81.2	52.3	71.5	100.0	136.3	85.4	72.3	100.0
3-shot$_{\text{CD}}$	106.8	59.8	100.0	100.0	162.7	82.1	100.0	100.0
ReClaim-Unified$_{\text{UD}}$	98.9	64.0	98.1	100.0	157.5	94.1	96.7	100.0
ReClaim-Unified$_{\text{CD}}$	79.4	51.6	100.0	100.0	139.5	84.4	100.0	100.0
ReClaim w/IG$_{\text{UD}}$	103.2	74.6	98.7	100.0	163.0	98.8	97.4	100.0
ReClaim w/IG$_{\text{CD}}$	99.1	73.1	100.0	100.0	167.4	100.5	100.0	100.0

Table 4: The generated text length, consistency of references, and proportion of attributed answer sentences in different methods. CD indicates the use of constrained decoding, while UD indicates the absence of constrained decoding.

Claim$_{\text{Unified}}$ method, compared to the 3-shot prompting method, can improve the fluency of model responses (15.6% on average), it has a minimal effect on correctness (1.6% on average) and significantly reduces citation quality (23.4% on average).

This indicates that using the ReClaim$_{\text{Unified}}$ method to fine-tune the model and generate responses in one step is almost ineffective for our tasks. Furthermore, it significantly harms our key metric, CAS, which is critical for the verifiability and credibility of the model's responses. We hypothesize the main reason as follows: generating the claim part only requires information from the previous reference part. However, excessive additional information in the input context likely disrupts this statement generation process. This extraneous information significantly interfered with the model's ability to learn to accurately generate claim based solely on the preceding reference part during the training process, thereby reducing its performance.

ReClaim w/IG Improves Attribution Experimental results indicate that the ReClaim w/IG method outperforms other methods in two citation quality metrics while maintaining high fluency and correctness scores. Specifically, when using the Llama3-8B-Instruct model, our method outperforms the ALCE method with ChatGPT by 23.7%, 30.0%, and 7.9% in fluency, CAS, and CRS metrics, respectively, and only shows a 4.0% decrease in correctness.

Compared to the ReClaim$_{\text{Unified}}$ method, the ReClaim w/IG approach's biggest difference lies in the training and inference strategies during the claim generation phase. It filters out extraneous contextual information and trains the model to generate claims based solely on the preceding reference part. The significant improvements in citation quality demonstrate the effectiveness of the strategy adopted by the ReClaim w/IG method.

As shown in Table 3, the experimental results demonstrate that the ReClaim w/IG method achieves the best performance in metrics of fluency and citation quality when compared to fine-tuning only the RefModel or the ClaimModel methods. This indicates the effectiveness of our fine-tuned dual-model approach. While the Claim-only method, which fine-tunes only the ClaimModel, yields the highest average answer correctness score (37.8), it comes at the cost of increased response length and reduced fluency. Additionally, numerous experiments indicate that improved citation quality often leads to a slight decrease in answer correctness. This is expected, as better attribution tends to constrain the source of information for model responses to the selected citations.

5.2 Can ReClaim Enhance the Verifiability and Credibility of RAG-based Question Answering?

To evaluate the verifiability of model responses and the impact of decoding constraints on their credibility, we measured the average length of citations, the consistency of citation texts (CR), and the proportion of sentences with attribution (AR) under different methods when using constrained decoding (CD) and unconstrained decoding (UD). The experimental results are shown in Table 4.

ReClaim Improves Verifiability Experimental results indicate that ReClaim's average citations length is only about 20% of the length produced by the ALCE method, significantly reducing the fact-checking time cost. Compared to the ALCE method, our method ensures specific attribution for each response sentence, further improving the verifiability of model's answer.

Decoding Constraints Improve Credibility Experimental results indicate that, compared to Prompting method, the fine-tuned ReClaim$_{\texttt{Unified}}$ and ReClaim w/IG method can improve the consistency of the reference text. However, it still does not ensure complete alignment with the source passages. By applying decoding constraints during the model's inference process, we ensure that each reference part is composed entirely of exact sentences from the source reference passages. This enhances the accuracy of the references and reduces the occurrence of hallucinations. Combining decoding constraints with our attributed text generation significantly improves the credibility of the model's responses.

6 Conclusion

We propose a attributed text generation method, ReClaim, which adds sentence-level fine-grained citations to model-generated answer in RAG systems. This approach alternates between generating citations and answer sentences through Prompting or by fine-tuning the model.

The results show that our method improves citation quality while maintaining answer quality compared to the baseline method. Additionally, our approach significantly reduces the length of citations, thus decreasing the time cost required for fact-checking and further enhancing the verifiability of the model's responses. Moreover, by using constrained decoding during citation generation, we ensure that each citation is composed of exact sentences from the source passages.

Although this paper focuses on long-form question answering task, we hope that our method can be extended to other tasks such as short-form question answering and Multi-document Summarization to improve the verifiability and credibility of model-generated responses.

Limitations

In this paper, our training dataset was exclusively targeted at long-form question-answering task, which reduces the generalization ability of our fine-tuning methods. Additionally, the construction of our training dataset did not account for the influence of irrelevant passages in the context on the model's ability to find citations. This may weaken the model's capability to filter out irrelevant information.

While our approach allows the model to synthesize information from multiple reference sentences for attribution, we did not specifically enhance this capability in our training data construction process and the model's inference process.

Ethics Statement

We hereby acknowledge that all authors of this work are aware of the provided ACL Code of Ethics and honor the code of conduct.

Datasets Source All original datasets used for training and testing were sourced from open and publicly accessible resources, and they are all approved for use in research purposes, thereby minimizing the risk of sensitive information leakage.

While we employed LLMs for automated processing during the construction of training dataset, data cleansing was performed to prevent the introduction of additional noise and bias. We solely utilize the constructed dataset for model training. Although paragraph retrieval is not the focus of our work, information retrieved from large corpora may introduce some noise and bias into the responses generated by LLMs. To address these issues, we will optimize the data construction process and explore methods for retrieving noise-free and unbiased information.

AI assistants AI assistants (ChatGPT) were solely used to improve the grammatical structure of the text.

References

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*.

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*, 33:1877–1901.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. Eli5: Long form question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3558–3567.

Edward Fredkin. 1960. Trie memory. *Communications of the ACM*, 3(9):490–499.

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

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*, pages 6465–6488.

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*, pages 3905–3920.

Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. 2021. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Gautier Izacard and Edouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. In *EACL 2021-16th Conference of the European Chapter of the Association for Computational Linguistics*, pages 874–880. Association for Computational Linguistics.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611.

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

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

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*, pages 4549–4560.

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. *arXiv preprint arXiv:2203.11147*.

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*.

OpenAI. 2022. Openai: Introducing chatgpt.

OpenAI. 2023. Gpt-4 technical report.

Krishna Pillutla, Swabha Swayamdipta, Rowan Zellers, John Thickstun, Sean Welleck, Yejin Choi, and Zaid Harchaoui. 2021. Mauve: Measuring the gap between neural text and human text using divergence frontiers. *Advances in Neural Information Processing Systems*, 34:4816–4828.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392.

Ivan Stelmakh, Yi Luan, Bhuwan Dhingra, and Ming-Wei Chang. 2022. Asqa: Factoid questions meet long-form answers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8273–8288.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. 2022. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.