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

Sirui Xia[1], **Xintao Wang**[1], **Jiaqing Liang**[2*], **Yifei Zhang**[1], **Weikang Zhou**[3]
Jiaji Deng[3], **Fei Yu**[3], **Yanghua Xiao**[1*]
[1]Shanghai Key Laboratory of Data Science, School of Computer Science, Fudan University
[2]School of Data Science, Fudan University [3]AntGroup
{srxia24, xtwang21, yifeizhang23}@m.fudan.edu.cn,
{liangjiaqing, shawyh}@fudan.edu.cn,
feiyu.fyyu@gmail.com, {zhouweikang.zwk, dengjiaji.djj}@antgroup.com

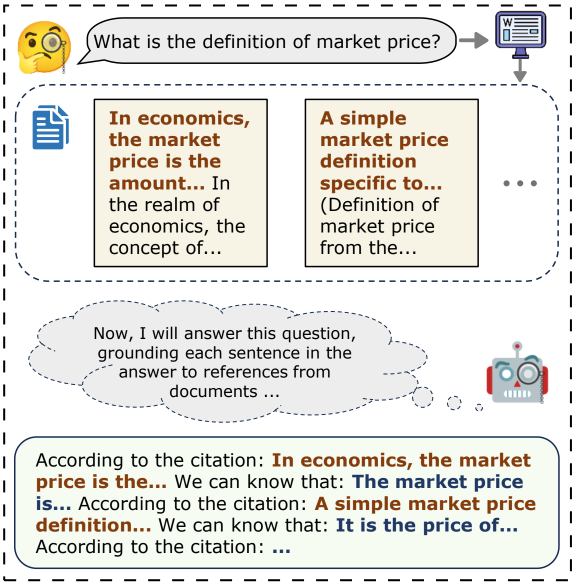
Abstract

Retrieval-Augmented Generation (RAG) has been widely adopted to enhance Large Language Models (LLMs) in knowledge-intensive tasks. To enhance credibility and verifiability in RAG systems, Attributed Text Generation (ATG) is proposed, which provides citations to retrieval knowledge in LLM-generated responses. Prior methods mainly adopt coarse-grained attributions, with passage-level or paragraph-level references or citations, which fall short in verifiability. This paper proposes ReClaim (**Re**fer & **Claim**), a fine-grained ATG method that alternates the generation of references and answers step by step. Different from previous coarse-grained attribution, ReClaim provides sentence-level citations in long-form question-answering tasks. With extensive experiments, we verify the effectiveness of ReClaim in extensive settings, achieving a citation accuracy rate of 90%.[1]

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 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 required 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...

...

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: **...**

Figure 1: The task setup for ReClaim. Given question and reference passages from a large corpus. The LLM then generates a response with fine-grained citations. For detailed examples, see Table 11.

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 in long-form question-answering using the RAG system.

To enhance the LLM's performance in attributed text generation, we developed a training dataset and fine-tuned the LLM to facilitate sentence-level citation selection from given reference passages and subsequent answer generation. We implemented an alternating strategy between citation generation and answer text generation. We apply constrained decoding during LLM inference by encoding reference passages into a token-level prefix tree. This

[1] Code and datasets are public at: https://github.com/pdxthree/ReClaim

Findings of the Association for Computational Linguistics: NAACL 2025, pages 969–988
April 29 - May 4, 2025 ©2025 Association for Computational Linguistics

restricts the LLM to generate citations only along the tree's paths, ensuring alignment with the reference passages and avoiding inconsistencies.

The results of experiments demonstrate that ReClaim outperforms existing baselines. ReClaim significantly improves the citation quality, enabling the citations to better support the answer. Furthermore, ReClaim greatly reduces the verbosity of citations, thereby easing the fact-checking process.

Our contributions are summarized as follows:

1. We propose a method called ReClaim, which alternately generates citations and answer sentences, enabling LLM to produce answers with sentence-level citations, thus enhancing the LLM's verifiability and credibility.

2. To enhance LLMs in sentence-level citation generation, we construct a dataset based on WebGLM-QA (Liu et al., 2023) and ELI5 (Fan et al., 2019) dataset. Then, we fine-tune Llama3-8B-Instruct (Dubey et al., 2024) models for reference and claim generation respectively, achieving better citation quality compared to the baseline method with ChatGPT (OpenAI, 2022).

3. Through extensive experiments, we demonstrate the effectiveness of our method in enhancing the LLM's verifiability and credibility, achieving performance comparable to much bigger models like ChatGPT.

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 generates comprehensive answers that require deep contextual understanding and information integration from multiple sources (Fan et al., 2019; Stelmakh et al., 2022; Malaviya et al., 2024).

Attributed Text Generation Many current works propose various methods for generating answer text with citations, differing in their approaches to attribution and citation granularity.

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. Additionally, some work has focused on fine-tuning models to improve the generation of attributed answer text (Huang et al., 2024a; Asai et al., 2023; Huang et al., 2024c).

In addition to the aforementioned methods that add citations directly during answer generation, there are some works that focus on finding citations afterward (Gao et al., 2023a). Some research further achieves better attribution performance through multiple retrievals and validations (Li et al., 2024; Sun et al., 2023).

3 Method

We introduce ReClaim, which aims to generate text with sentence-level citations. Specifically, ReClaim generates citations with answers alternatively, in a step-by-step manner. We first introduce the overview of ReClaim in Section 3.1, and then detail the specific implementation in Section 3.2 to 3.5.

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 LLM 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 (provided as text, not just source numbers) and n claims $c_1, ..., c_n$, which are portions of the

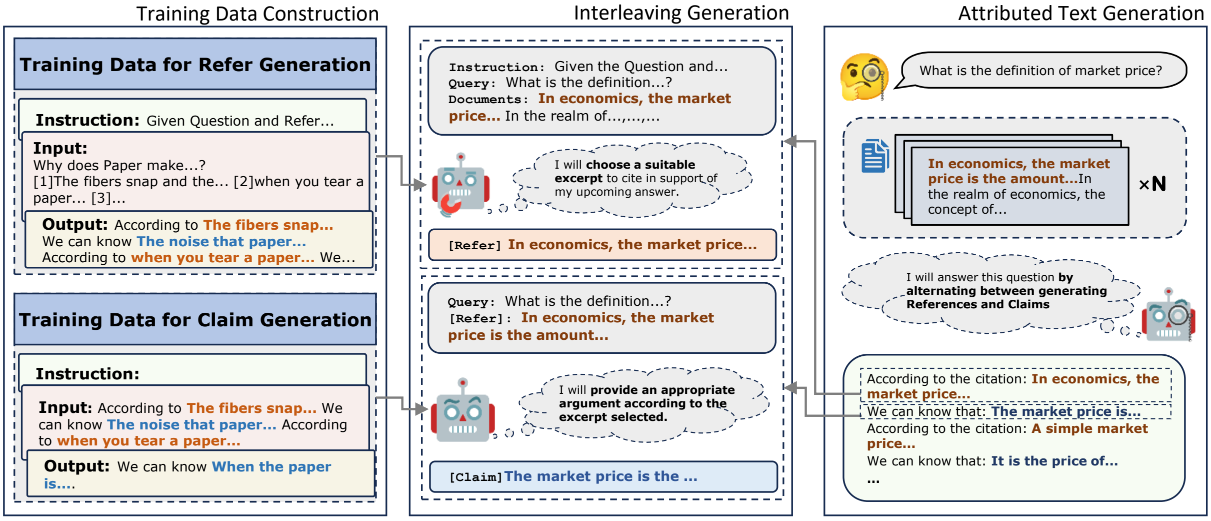

Training Data Construction
Training Data for Refer Generation
Instruction: Given Question and Refer...
Input:
Why does Paper make...?
[1]The fibers snap and the... [2]when you tear a paper... [3]...
Output: According to The fibers snap...
We can know The noise that paper...
According to when you tear a paper... We...
Training Data for Claim Generation
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....
Interleaving Generation
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 ...
Attributed Text Generation
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 LLM 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 base model, which alternately switches between the fine-tuned models and the input context during inference.

LLM's response generated based on these references. Each reference r_i serves as a substantiation of claim c_i, and together, all c_i form the LLM's complete answer to the question.

During generation, the LLM alternates between generating references and claims. However, experiments revealed that the LLM faces several issues: 1) The generated references are not always consistent with the retrieved passages; 2) The generated claims do not always attribute well to the corresponding references. Therefore, in the following sections, we study how to improve the generation of references and claims.

3.2 Training Dataset Construction

To improve the LLM's ability to choose the references and generate corresponding claims, we construct a specialized fine-tuning dataset based on the WebGLM-QA (Liu et al., 2023) and ELI5 (Fan et al., 2019) datasets.

The training data input includes an instruction, a query, and reference passages. The output consists of multiple reference-claim pairs, with the reference being a sentence-level citation and the claim being a part of the answer. The reference and claim components are designed to train the LLM's ability to select accurate and relevant fine-grained citations from the complete reference passages, and to generate fluent, coherent answer texts that are faithful to the selected citations.

The steps of our dataset construction are detailed as follows:

Reference Passages Retrieval In the WebGLM-QA Dataset, the reference passages are fine-grained texts closely aligned with the query. However, some research has shown that irrelevant reference texts significantly impact the quality of LLM-generated answers (Wu et al., 2024). To enhance the LLM's performance against irrelevant contexts, we sample a portion of the constructed WebGLM-QA training dataset, retrieve the top-100 passages using BM25, and calculate the similarity between the query and passages with a Reranker model (Chen et al., 2024). Passages with high BM25 ranking but low re-rank score are selected as irrelevant text and added to the training data.

For the ELI5 dataset, we use BM25 to retrieve the top-100 reference passages, which included some irrelevant noise to simulate real-world retrieval conditions. Additionally, we use the Reranker to re-rank the top 100 passages retrieved from a subset of the selected ELI5 dataset, producing a refined re-ranked dataset.

Model Answer Generation For the WebGLM-QA dataset, we directly use the original model responses. For the ELI5 datasets, due to significant information discrepancies between the human answers and the retrieved passages, we provide the top-5 retrieved passages to the LLM (Llama-3.1-405B-Instruct (Dubey et al., 2024)) to generate long-form answers.

Multi-Stage Citation Search To select sentence-level fine-grained citations for the answers generated by the LLM, we employ a multi-stage citation search approach that moves from coarse to fine.

We first enable the Llama-3.1-405B-Instruct model to automatically segment the text into clauses. For each clause, the model identifies the minimal set of citations from the reference passages that sufficiently support it.

Then, we use NLI (Natural Language Inference) model (Honovich et al., 2022) to evaluate the entailment relationship between the chosen citations (as the premise) and the corresponding clause (as the hypothesis). To enhance filtering precision, we set a threshold ($\theta = 0.8$) to discard cases where the entailment probability between the chosen citation and the clause falls below the threshold.

Dataset	Samples	Average Length		
		Answer	Citation	Passages
WebGLM-QA	4582	98.53	154.62	326.57
WebGLM-QA Extend	2605	83.85	114.1	396.16
ELI5 Default	3383	91.33	132.34	545.01
ELI5 Rerank	2653	107.54	158.52	542.02

Table 1: Statistics of the training dataset. For more details, please refer to the appendix B.

3.3 Unified Generation

The ReClaim `Unified` method uses a simple fine-tuning and inference approach. It first performs instruction fine-tuning on the LLM using the dataset constructed in Section 3.2. Then, it uses the fine-tuned LLM 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 LLM 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 method trains separate LLMs for the generation of the reference parts and the claim parts, and alternates between the two LLMs during answer generation, adjusting the input to each LLM accordingly (IG represents the interleaving use of two fine-tuned LLMs for the iterative generation of references and claims).

The entire generation process, as illustrated in Figure 2, involves the following steps to train the LLMs and generate the answer.

Reference Generation During the generation of reference parts, the LLM 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 LLM for generating the reference parts does not require masking parts of the input context information, we follow the same approach as in the Section 3.3 to fine-tune the ReferModel.

Claim Generation During the generation of claim parts, the LLM 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 utilize the training dataset from Section 3.2, formatting it to align with our ClaimGen generation process, and then use this formatted dataset to fine-tune the ClaimModel.

3.5 Decoding Constraints

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

Sentence Segmentation and Encoding We segment the reference passages into individual sentences. Then, we use the LLM 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 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 LLM 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 LLMs 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

ASQA We evaluate the 948 samples from the ASQA dataset (Stelmakh et al., 2022), selected by ALCE, and use the five oracle paragraphs provided by ALCE as reference passages, which are chosen from the top 100 retrieved passages representing the gold passages.

ELI5 We evaluate the 1,000 samples selected from ELI5 dataset (Fan et al., 2019) by ALCE and use the five oracle passages as reference passages.

EXPERTQA To test the LLM's generalization ability under the standard RAG process, we select 1,000 samples from the EXPRTQA (Malaviya et al., 2024) dataset. We follow the standard RAG procedure: using BM25 to retrieve the top-100 passages, then re-ranking them to select the top 5 passages.

4.2 Evaluation Metrics

Building on our previous task definition, we focus on evaluating the LLM-generated outputs in three key areas. Below, we introduce three evaluation dimensions, with more detailed information and calculation methods provided in Appendix C.3.

Answer Quality 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 and EXPERTQA 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 measure 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 whether the answer is entirely supported by cited sentences and is the most critical metric in our evaluation; 2) Citation Redundancy Score (CRS), which identifies any redundant citation sentences.

We use the NLI (Honovich et al., 2022) model to compute the entailment relationship between each reference part and the corresponding claim part, and the final CAS score is the proportion of correctly attributed sentences in the answer.

We also use the NLI model need to determine if the reference contains redundant sentences, and the final CRS score is the proportion of non-redundant sentences relative to all sentences in the references.

Verifiability We employ three 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; 3) Consistency Ratio (CR), which represents the proportion of text consistency between the reference parts and the reference passages through string matching.

4.3 Baselines

We use only methods that restrict citation sources to the top-k reference passages as our comparison baselines. We exclude methods that introduce additional retrieval steps, such as VTG (Sun et al., 2023) and LLatrieval (Li et al., 2024), from our comparison.

Prompting-Based Following the ALCE (Gao et al., 2023b) method, we prompt ChatGPT and Llama3.1-8B-Instruct with few-shot demonstrations that consist of a query, the top-5 retrieved passages, and an answer with inline citations. Additionally, we perform four samples of the generation from Llama3.1-8B-Instruct and select the best answer from each case as the experimental result for the w/Rerank method.

Post-hoc Following the ALCE method, we generate answer using the Llama3-8B-Instruct model

Method	Model	ASQA				ELI5				EXPERTQA			
		Fluency	Correct.	Citation		Fluency	Correct.	Citation		Fluency	Correct.	Citation	
		MAUVE	EM Rec.	CAS	CRS	MAUVE	Claim Rec.	CAS	CRS	MAUVE	Claim Rec.	CAS	CRS
Prompting-Based													
ALCE	ChatGPT	64.4	48.9	74.5	72.7	59.4	**21.3**	57.8	56.0	53.7	20.5	66.7	64.9
	Llama3-8B	79.2	55.2	54.7	54.6	45.4	20.5	42.8	39.3	56.7	20.2	51.4	47.8
	w/Rerank	81.3	55.3	79.1	76.4	37.6	20.0	59.5	53.8	45.3	19.8	68.7	60.2
Post-hoc													
ClosedBook	Llama3-8B	50.5	28.9	13.7	13.7	66.8	16.8	17.4	17.4	37.6	**25.4**	11.2	11.2
Open-book	Llama3-8B	73.7	53.2	52.1	52.1	26.6	20.8	31.9	32.0	58.7	21.0	36.9	36.8
Training-Based													
Self-RAG	Llama2-7B	70.6	38.7	53.3	66.2	33.1	9.7	23.1	33.9	19.0	12.5	9.4	12.1
RS+RL	Llama2-7B	84.4	47.7	75.5	69.4	43.6	19.1	59.1	51.6	46.6	15.3	67.8	60.1
FRONT	Llama2-7B	72.5	56.5	72.2	66.0	49.7	18.1	64.0	59.1	56.5	14.7	73.6	**68.9**
Our Methods													
0-shot	GPT-4o	72.9	52.8	74.8	51.6	37.3	19.9	63.5	30.7	59.4	18.5	73.4	26.5
3-shot	Llama3-8B	90.1	50.7	77.7	62.1	61.3	17.9	78.3	45.3	46.0	12.6	79.8	51.4
	ChatGPT	74.9	52.6	72.5	63.4	27.8	17.8	68.6	50.8	30.6	14.1	72.7	55.7
	GPT-4o	**91.3**	**56.6**	77.4	58.0	29.7	21.1	70.2	36.8	38.7	18.2	68.0	45.2
ReClaim Unified	Llama3-8B	89.8	53.3	68.2	58.9	**73.6**	19.9	69.4	48.6	64.2	16.4	70.0	50.4
ReClaim w/IG	Llama2-7B	71.4	55.0	89.5	78.7	67.3	17.6	86.3	58.6	**67.2**	12.7	86.3	60.2
	Llama3-8B	88.1	53.5	**92.1**	**86.1**	71.6	17.8	**89.9**	**67.5**	63.5	14.0	**90.1**	68.6

Table 2: Results on ASQA, ELI5, EXPERTQA. Definitions for Fluency, Correct. and Citation are in Section 4.2.

with the closed-book approach based on the given query. Then, for each statement in the answer, we search the top-100 passages for citations.

We also provide the LLMs with the top-5 reference passages, allowing it to generate the answer based on them and then re-identify citations within the top-5 passages. We refer to this method as open-book.

Training-Based Following Self-RAG (Asai et al., 2023), the LLM is trained to retrieve passages on demand, verify relevance, and generate answers based on the retrieved content. We directly provide the selfrag-7B (fine-tuned Llama2-7B model) with the top-5 passages for generation.

Following the method in (Huang et al., 2024a), we use the LLM trained with fine-grained rewards (RS+RL, fine-tuned Llama2-7B model) to generate answer with citations. Additionally, we compare the fine-grained attribution method FRONT (Huang et al., 2024b), which is fine-tuned based on the Llama2-7B model.

4.4 Methods

We evaluate ReClaim by testing several LLMs with different generation approaches.

ReClaim prompting We directly prompt LLMs to generate answer with citations, using GPT models (OpenAI, 2023, 2022) and the Llama series (Touvron et al., 2023) to evaluate the performance and effectiveness.

ReClaim Unified We follow the methodology in Section 3.3 and use the training dataset constructed in Section 3.2 to fully fine-tune the Llama3-8B-Instruct model, then conduct one-step generation with citations to evaluate the effectiveness of the ReClaim Unified method.

ReClaim w/IG As outlined in Section 3.4, we fine-tune the same base model (Llama2-7B-hf and Llama3-8B-Instruct), generating two separate LLMs. During the inference phase, we alternate between these two LLMs to interleavingly generate the reference and claim parts.

For these two fine-tuning methods, we adopt the constrained decoding described in Section 3.5 to limit the generation of reference parts. For more experimental details, see Appendix C.

4.5 Ablation Study

We primarily conduct ablation experiments on the ReClaim w/IG method to investigate the necessity of training data filtering and fine-tuning the ReferModel and ClaimModel. All fine-tuned models are based on the Llama3-8B-Instruct model.

Method	ASQA				ELI5				EXPERTQA			
	Length		Consistency	Attri.	Length		Consistency	Attri.	Length		Consistency	Attri.
	Citation	Claim	CR	AR	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	627.5	115.1	100.0	84.1
RS+RL	327.0	39.9	100.0	94.5	476.6	80.8	100.0	93.2	501.9	81.9	100.0	95.1
RECLAIM 3-shot	106.8	59.8	75.5	100.0	162.7	82.1	72.1	100.0	198.7	82.1	77.5	100.0
RECLAIM Unified	77.9	52.9	100.0	100.0	139.8	93.1	100.0	100.0	150.9	99.2	100.0	100.0
RECLAIM w/IG	82.8	68.9	100.0	100.0	145.0	104.7	100.0	100.0	157.5	109.1	100.0	100.0

Table 3: The generated text length, consistency of references, and proportion of attributed answer sentences in different methods. RECLAIM 3-shot does not use decoding constraints, while RECLAIM Unified and RECLAIM w/IG use decoding constraints.

Method	ASQA				ELI5				EXPERTQA			
	MAUVE	EM Rec.	CAS	CRS	MAUVE	Claim Rec.	CAS	CRS	MAUVE	Claim Rec.	CAS	CRS
ReferModel-Only w/Extend	29.4	29.8	70.5	58.9	<u>61.4</u>	13.4	59.9	40.6	<u>50.0</u>	10.1	61.7	44.4
ReferModel-Only w/Sum	29.2	33.0	**94.5**	<u>80.8</u>	46.8	14.6	**96.2**	<u>61.4</u>	49.5	11.0	**97.7**	**69.1**
ClaimModel-Only	<u>43.6</u>	**57.7**	89.6	77.2	54.5	**18.9**	84.4	55.1	43.2	**14.8**	85.6	57.2
RECLAIM w/IG	**88.1**	<u>53.5</u>	<u>92.1</u>	**86.1**	**71.6**	<u>17.8</u>	<u>89.9</u>	**67.5**	**63.5**	<u>14.0</u>	<u>90.1</u>	<u>68.6</u>

Table 4: Ablation study results. The underline indicates the second largest value.

Pre-Filtered Training Data To investigate the necessity of using an NLI model for training data filtering, we fine-tune the LLM on the pre-filtered data while keeping the dataset size and training parameters unchanged. The experimental results are shown in Figure 3.

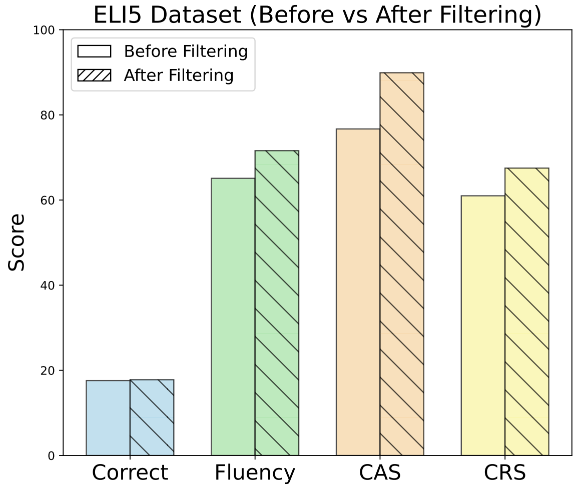


Figure 3: Comparison of the performance of LLMs trained on the training dataset before and after filtering on the ELI5 dataset. The comparative experimental results of the ASQA and EXPERTQA datasets are presented in Appendix Figure 9 and Figure 10.

The experimental results show that fine-tuning the LLM with data filtered by the NLI model improves its performance across various evaluation metrics.

Fine-tuned ReferModel Only We use the fine-tuned ReferModel to generate reference parts and the base model to generate claim parts. When generating claims, it is essential to ensure that it adheres to the information in the preceding reference and maintains fluency with the previously generated claims. We explore two prompt strategies:

1. **Extension**: We provide the LLM with the previous reference and all preceding claims, requiring the LLM to extend the claims based on the information in the reference.

2. **Summary**: We provide the LLM with a reference so that it can directly generate a brief summary based on the previous reference.

Fine-tuned ClaimModel Only We employ the 3-shot prompting approach to generate the reference section using the base model and utilize the fine-tuned ClaimModel for generating the claim section.

The experimental results are shown in Table 4.

4.6 Faithfulness Analysis

To further validate our approach in improving the credibility of model responses, we use GPT-4o-mini (OpenAI, 2023) as our evaluation model to evaluate the faithfulness metric of model answers against complete reference passages using the assessment method proposed by RAGAS (Es et al., 2023). This helps us determine whether the LLM's generated answers are fully based on the information in the given reference paragraphs.

We prioritize the CAS metric and use faithfulness as a supplementary measure to evaluate credibility improvements. Since faithfulness to a citation implies faithfulness to the original text, a high faithfulness score does not guarantee strong CAS performance, as an answer true to the full document may not align with the selected citation.

We evaluate the faithfulness metric on the test datasets, and the results for the ELI5 dataset are shown in Figure 4. Results for other datasets are shown in Appendix Figure 7 and 8.

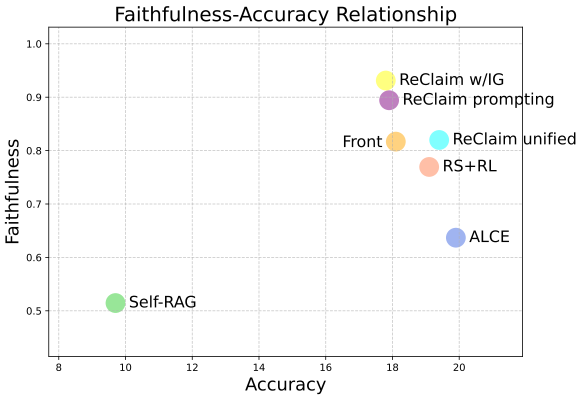


Figure 4: The x-axis represents the accuracy of the LLM's responses, while the y-axis shows the faithfulness score. For the Self-RAG and RS+RL methods, we use the fine-tuned 7B model, whereas for other methods, we employ Llama3-8B-Instruction as the base model.

Our method achieved the highest faithfulness score. This demonstrates that while our approach may slightly reduce answer quality, it significantly enhances answer faithfulness and minimizes unnecessary hallucinations linked to the LLM's internal parameters.

The results show an inverse relationship between the accuracy of LLM-generated answers and their faithfulness to reference passages. Higher faithfulness and more granular citations have narrowed the scope of our answer sources, which may contribute to the lower accuracy in LFQA task.

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.

ReClaim prompting Works Experimental results show that directly prompting LLMs yields satisfactory outcomes. Our approach improves average answer fluency and citation accuracy (CAS) compared to baseline methods. Notably, Llama3-8B-Instruct surpasses other baselines in CAS scores, including ALCE+ChatGPT.

Although our method performs worse in CRS, the finer granularity of our citations reduces the impact of redundant content. Redundant citations only add a single irrelevant sentence, which does not significantly increase the cost of fact-checking.

ReClaim Unified Cannot Improve ATG Experimental results show that the ReClaim Unified method significantly reduces citation quality (CAS). This indicates that fine-tuning the LLM in this way fails to teach it how to generate claims based solely on the information from the previous reference.

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.

Compared to ALCE using ChatGPT, our method (using Llama3-8B) shows an average improvement of 31.3% on the CAS, 16.7% on the CRS, and 25.7% on the MAUVE across three test datasets, with only a 6.0% decrease in answer accuracy.

Specifically, we achieved an average CAS score of 90.7 across three test datasets, which is a crucial metric for assessing the degree of text attribution.

Compared to the ReClaim 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 LLM 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 4, the results of the ablation experiments indicate that fine-tuning two LLMs for alternating generation of references and claims achieves the most balanced performance.

While ReferModel-Only w/Sum method yields a high citation quality score, it compromises the accuracy and fluency of the answers.

On the other hand, using ClaimModel-Only method for generation achieves higher accuracy scores, but it negatively affects answer fluency, and the generation often tends to produce excessively long answers. Additionally, to ensure that the base model adheres to our format when generating reference parts, we need to provide multiple examples, which reduces the effective context window length.

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

The results of the average length of citations, citation consistency (CR), and attribution ratio (AR) are shown in Table 3.

ReClaim Improves Verifiability Experimental results show that ReClaim's average citation length is about 22% of the ALCE method, significantly reducing fact-checking time.

ReClaim ensures that each response sentence is supported by a specific citation source. By employing constrained decoding, our method ensures that every citation is directly extracted from the original text, significantly reducing the occurrence of hallucinations during the citation generation process. Together, these measures collectively enhance the verifiability of the generated answers.

ReClaim Improve Credibility As mentioned in Section 5.1 regarding the improvement of attribution, this enhancement boosts the credibility of the LLM's responses by providing a definitive source of evidence for the generated answers.

In addition, as shown in Figure 4, our method increases the faithfulness of the LLM's responses to the overall reference passages by confining the sources of response information within the chosen citations. This improvement reduces the influence of the LLM's internal knowledge on answer generation, thereby enhancing the credibility of our approach.

Our method strictly generates answers based on reference passages, aiming to improve citation quality in LLM responses through stringent constraints. The accuracy of our answers depends heavily on the information density of the reference passages and cannot rely on the LLM's internal knowledge, which may also explain the slight decrease in answer accuracy in our approach.

6 Conclusion

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

The results show that our method improves citation quality while maintaining answer quality compared to the baseline methods. 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 LLM's responses. Moreover, by using constrained decoding during citation generation, we ensure that each citation is composed of exact sentences from the source passages.

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, our method requires explicitly outputting the cited sentences, which often leads to generating answers that are more than double the length. This results in increased output time for the LLM.

On one hand, while our approach allows the LLM to synthesize information from multiple reference sentences for attribution, we did not specifically enhance this capability during the training data construction and LLM inference processes. On the other hand, our answer template has certain limitations, as not every response generated by the LLM requires a citation from the reference passages. Therefore, our method has limitations in scenarios that require synthesizing information from multiple sources or necessitate multi-hop reasoning to draw conclusions.

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, the retrieved information from large corpora may introduce noise and bias into LLM-generated responses. 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

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2023. Self-rag: Learning to retrieve, generate, and critique through self-reflection. *arXiv preprint arXiv:2310.11511*.

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*.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *arXiv preprint arXiv:2402.03216*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Shahul Es, Jithin James, Luis Espinosa-Anke, and Steven Schockaert. 2023. Ragas: Automated evaluation of retrieval augmented generation. *arXiv preprint arXiv:2309.15217*.

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.

Constanza Fierro, Reinald Kim Amplayo, Fantine Huot, Nicola De Cao, Joshua Maynez, Shashi Narayan, and Mirella Lapata. 2024. Learning to plan and generate text with citations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11397–11417, Bangkok, Thailand. Association for Computational Linguistics.

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.

Chengyu Huang, Zeqiu Wu, Yushi Hu, and Wenya Wang. 2024a. Training language models to generate text with citations via fine-grained rewards. *arXiv preprint arXiv:2402.04315*.

Lei Huang, Xiaocheng Feng, Weitao Ma, Yuxuan Gu, Weihong Zhong, Xiachong Feng, Weijiang Yu, Weihua Peng, Duyu Tang, Dandan Tu, and Bing Qin. 2024b. Learning fine-grained grounded citations for attributed large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 14095–14113, Bangkok, Thailand. Association for Computational Linguistics.

Lei Huang, Xiaocheng Feng, Weitao Ma, Yuxuan Gu, Weihong Zhong, Xiachong Feng, Weijiang Yu, Weihua Peng, Duyu Tang, Dandan Tu, et al. 2024c. Learning fine-grained grounded citations for attributed large language models. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 14095–14113.

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.

Chaofan Li, Zheng Liu, Shitao Xiao, and Yingxia Shao. 2023. Making large language models a better foundation for dense retrieval.

Xiaonan Li, Changtai Zhu, Linyang Li, Zhangyue Yin, Tianxiang Sun, and Xipeng Qiu. 2024. Llatrieval: Llm-verified retrieval for verifiable generation. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5453–5471.

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.

Chaitanya Malaviya, Subin Lee, Sihao Chen, Elizabeth Sieber, Mark Yatskar, and Dan Roth. 2024. Expertqa: Expert-curated questions and attributed answers. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3025–3045.

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.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Dmytro Okhonko, Samuel Broscheit, Gautier Izacard, Patrick Lewis, Barlas Oğuz, Edouard Grave, Wen-tau Yih, et al. 2021. The web is your oyster-knowledge-intensive nlp against a very large web corpus. *arXiv preprint arXiv:2112.09924*.

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.

Hao Sun, Hengyi Cai, Bo Wang, Yingyan Hou, Xiaochi Wei, Shuaiqiang Wang, Yan Zhang, and Dawei Yin. 2023. Towards verifiable text generation with evolving memory and self-reflection. *arXiv preprint arXiv:2312.09075*.

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*.

Siye Wu, Jian Xie, Jiangjie Chen, Tinghui Zhu, Kai Zhang, and Yanghua Xiao. 2024. How easily do irrelevant inputs skew the responses of large language models? In *First Conference on Language Modeling*.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, Zhangchi Feng, and Yongqiang Ma. 2024. Llamafactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, Bangkok, Thailand. Association for Computational Linguistics.

A Notation Table

	Definition
	Task Formulation
q	The given question.
$\mathcal{D}$	A set of reference passages, where $d \in \mathcal{D}$ is a passage.
$\mathcal{O}$	The response generated based on the question q and passages $\mathcal{D}$, composed of $\mathcal{O} = \{r_1, c_1, ..., r_i, c_i\}$
r	A portion of the citation, comprising certain sentences from $\mathcal{D}$.
c	A portion of the answer, formed by concatenating all c in sequence as a response to q.
	Methods
ALCE	A benchmark for Automatic LLMs' Citation Evaluation.
ReClaim-Unified	End-to-end data training and one-step generation.
ReClaim w/IG	Independent model training and Interleaving Generation.
Citation-only	Use the fine-tuned model to generate references and the base model with 3-shot prompting to generate claims.
Claim-only	Use the base model with 3-shot prompting to generate references and the fine-tuned model to generate claims.
	Metrics
MAUVE	Measuring the gap between neural text and human text using divergence frontiers.
EM Rec.	Exact match recall rate of gold short answers in the text generated by the LLM.
Claim Rec.	Recall rate of generated claims in the text generated by the LLM.
CAS	Correct attribution score, the proportion of sentences predicted as correctly attributed among all the sentences in the answer.
CRS	Citation redundancy score, the proportion of non-redundant sentences relative to all sentences in the references.
CR	Consistency ratio, the text consistency between the reference parts and the reference passages through string matching.
AR	Attribution ratio, the proportion of sentences in the output that are attributed.

Table 5: The notation table.

In Table 5, we list the notations and abbreviations in this paper, together with their definitions.

B Details of Training Dataset Construction

B.1 Details of Training Dataset

The statistics of the training dataset is shown in Table 1

WebGLM-QA The WebGLM-QA (Liu et al., 2023) dataset is a pioneering resource designed to bolster the development and assessment of web-enhanced question answering systems. It distinguishes itself by seamlessly integrating web search functionalities into the QA process, empowering systems to tap into the expansive repository of knowledge on the internet. Carefully curated to overcome the shortcomings of existing datasets, WebGLM-QA offers a holistic and pragmatic solution for open-domain question answering tasks. It consists of 43,579 high-quality data samples for the train split, 1,000 for the validation split, and 400 for the test split. Refer to our paper for the data construction details.

In WebGLM-QA, the provided answers is generated by the model based on the given reference passages; therefore, we directly use it as our gold answers.

ELI5 ELI5 (Fan et al., 2019) dataset is a benchmark in natural language processing designed for long-form question answering tasks, focusing on complex and explanatory questions (see Table 11). It comprises 270K threads from the Reddit forum "Explain Like I'm Five" (ELI5) where an online community provides answers to questions which are comprehensible by five year olds. For ALCE (Gao et al., 2023b), ELI5 questions were paired with passages from Sphere (Piktus et al., 2021), a filtered version of Common Crawl.

For the ELI5 dataset, due to the distributional differences between human answers and the retrieved reference passages, we opted to use Llama-3.1-405B-Instruct to generate answers based on the top-5 reference passages as the gold answers.

B.2 Citation Selection

During the training data structuring phase, we initially employ Llama-3.1-405B-Instruct to automatically segment the answers within the WebGLM-QA dataset and ELI5 dataset into clauses. Subsequently, for each of these clauses, we undertake a search for corresponding sentence-level, fine-grained citations from the provided reference passages. The prompt we use is as follows:

Upon obtaining answer clauses and their respective citations, structure the response in a cot format, as illustrated in Table 8.

Prompt 1: prompt for citation selection

You will be given reference passages and an answer.
Your task is to find a set of evidence for each answer clause from the reference passages.
You need to ensure that each evidence set is a combination of original sentences from the reference passages, and that any information in the corresponding clause must come entirely from the evidence set you identified.
Try to consolidate the answer content that is cited from the same continuous text into one clause.

Reference Passage:
{reference}
Original Answer:
{answer}
Output Format:
Clause: [Answer clause]
Evidence Set: [Combine sentences from the reference text that support this clause]
Clause: [Answer clause]
Evidence Set: [Combine sentences from the reference text that support this clause](Continue for all clauses)
...

B.3 Citation Filtering

After identifying the citation sentences corresponding to each answer clause, we need to filter out citation texts that diverge from the original reference passages or lack sufficient information to substantiate the answer clauses, the algorithm we use is as Algorithm 1.

C Experiment Settings

C.1 Details of Datasets

The statistic of the test datasets is shown in Table 9. In Figure 5 and Figure 6, we employ bge-v2-m3 (Li et al., 2023; Chen et al., 2024) to compute the relevance between queries and reference passages, presenting the statistical information of both training and test dataset in the form of box plots. Below, we provide a detailed description of the test datasets.

ASQA ASQA (Stelmakh et al., 2022) is the first long-form question answering dataset that focuses on ambiguous factoid questions (see Table 11). It contains 4,353 samples for the train split and 948 samples for the dev split. For ALCE (Gao et al., 2023b), ASQA questions were paired with passages from Wikipedia passages (2018-12-20 snapshot) which purportedly contained the answers.

Algorithm 1 Citation Filtering

```
Data: Question $q$, Reference passages $\mathcal{D}$, Answer $a = \{(r_i, c_i) \mid r_i \in R, c_i \in C\}$, NLI model $M$
for each $Tuple(q, \mathcal{D}, a)$ do
    $flag \leftarrow 1$
    for each citation $r_i$ and its corresponding clause $c_i$ do
        for $sentence \in r_i$ do
            if sentence not in $\mathcal{D}$ then
                $flag \leftarrow 0$
            end if
        end for
        $m \leftarrow p(M(r_i, c_i) = 1)$
        if $m < 0.8$ then
            $flag \leftarrow 0$
        end if
    end for
    if $flag = 0$ then
        remove $Tuple(q, \mathcal{D}, a)$
    end if
end for
```

EXPERTQA EXPERTQA (Malaviya et al., 2024) dataset is a benchmark designed for evaluating the factuality and attribution capabilities of large language models across diverse domains. It contains 2177 long-form questions curated by experts from 32 fields, along with LLM-generated answers that have been revised by these experts to ensure factuality and proper sourcing. The dataset aims to provide a high-quality resource for developing and assessing AI systems that can deliver accurate and well-referenced information tailored to the needs of domain specialists.

ELI5 Descriptions can be found in the section B.1

C.2 Details of Methods

Prompting In prompting setting, given a question and reference passages, we simply prompt the model to generate the attributed answer in a prescribed format. The prompt we use is as follows:

Prompt 2: prompt for ATG

Given the Question and References below, provide an answer for the Question that is generated using information exclusively from the References(some may be irrelevant).
Please use the format of:
According to the citation: <reference> {reason1} </reference> We can know that: <claim> {answer1} </claim> According to the citation: <reference> {reason2} </reference> We can know that: <claim> {answer2} </claim> According to the citation: <reference> {reason3} </reference> We can know that ...
The {reason} consists of one or more reference sentences in the References. The {answer} is generated based solely on the information contained within {reason}.
You may employ multiple such structures to organize your answer, ensuring that when all the {answer}s are concatenated, they maintain coherence, fluency, and collectively constitute a comprehensive response to the Question.
Strive to generate a longer text, utilizing several such structures to organize your response.

# Question: {question}
# References: {reference passages}
# Output:

ReClaim Unified In ReClaim Unified setting, we directly train the model using the constructed training dataset in Appendix B and employ the same instruction as in the prompting setting for one-step generation of attributed text.

ReClaim w/IG In ReClaim w/IG setting, we train two separate models for generating the reference and claim parts of attributed answers. Specifically, we train the model using the constructed training dataset in Appendix B as ReferModel for producing reference parts. For ClaimModel, which generates claim parts, we construct a new form of training data based on the original dataset. The detailed structure of this tailored training data is shown in Table 10. In Table 12, we provide an illustration of an interleaving generation instance.

C.3 Details of Evaluation Metrics

EM Rec. This metric is employed to assess the accuracy of the ASQA dataset. It utilizes a list of short answers (typically in the form of words or phrases) provided within the ASQA dataset to perform string matching within the generated long answers generated. The proportion of short answers that can be matched relative to the total length of the short answer list is calculated to determine the metric.

Claim Rec. This metric is designed to evaluate the accuracy of the ELI5 and EXPERTQA datasets. Since these datasets do not provide short answer lists but only a long-form answer, ALCE employs InstructGPT (Ouyang et al., 2022) to sample three sub-claims from the standard answer. Then use the NLI model to assess the entailment relationship between each sub-claim and the LLM-generated response. The final score for this metric is determined by the proportion of sub-claims that are entailed in the LLM's answer.

MAUVE We use MAUVE (Measuring the gap between neural text and human text using divergence frontiers) to measure the fluency of the generated text. MAUVE is a statistical metric that quantifies the similarity between neural-generated text and human-written text by computing the divergence frontiers between them. A higher MAUVE score indicates that the generated text is more coherent and natural, closely resembling human language style.

CAS This metric, consistent with the Citation Recall metric in ALCE, is designed to evaluate the proportion of answer clauses that are sufficiently supported by citations. Specifically, for each answer clause associated with a citation, a Natural Language Inference (NLI) model is employed to determine whether an entailment relationship exists between the clause and its corresponding citation. If such a relationship is identified, it indicates that the citation adequately substantiates the answer clause. The final score for this metric is derived from the ratio of answer clauses that exhibit an entailment relationship with their citations.

CRS This metric aligns with the Citation Precision metric in ALCE, designed to assess whether the citations contain redundancy. In the ALCE, citations are identified by paragraph numbers, and redundancy is evaluated by sequentially removing paragraphs. However, in our approach, to accommodate sentence-level citations, we first segment the citation into sentences using NLTK. We then iteratively remove sentences and verify whether the remaining text still maintains an entailment relationship with the answer clause. Finally, the proportion of non-redundant citations is used as the CRS score.

C.4 Training Details

We trained the model using llama-factory (Zheng et al., 2024) on four A800 80G GPUs. For the Unified approach, we applied full fine-tuning with a learning rate of 3e-5, 3 epochs, and a total batch size of 128. For the ReClaim w/IG approach, we used LoRA to train two models: ReferModel with a learning rate of 5e-5, 5 epochs, and a total batch size of 128, and ClaimModel with a learning rate of 3e-5, 2 epochs, and a total batch size of 128. During model inference, we can alternate between two LoRA parameters to reduce memory requirements.

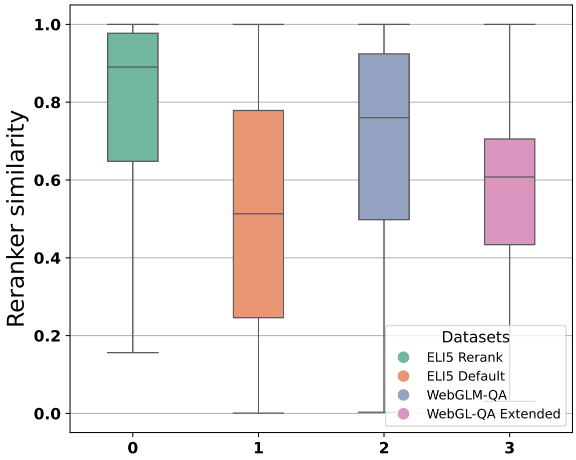


Figure 5: Reranker score distribution between query and reference passages in the training dataset.

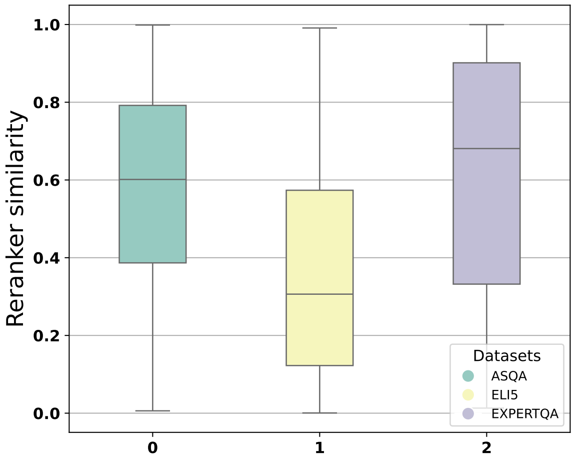


Figure 6: Reranker score distribution between query and reference passages in the test datasets.

D Additional Results

D.1 Default Datasets

For our test datasets, in addition to providing gold passages or reranked passages as reference passages, we also tested the results of directly using the top-5 retrieved passages, as shown in Table 6.

Dataset	Type	Fluency	Correct.	CAS	CRS
ASQA	Oracle	88.1	53.5	92.1	71.6
	Default	88.1	40.0	91.0	84.4
ELI5	Oracle	71.6	17.8	89.9	67.5
	Default	75.7	6.5	89.4	66.3
EXPERTQA	Rerank	63.5	14.0	90.1	68.6
	Default	62.9	12.2	87.2	65.7

Table 6: Results of the default test datasets.

The results reveal that although the accuracy of the LLM's responses decreases due to varying quality of reference passages provided to the LLMs, it still maintains high CAS and CRS scores. This indicates that our method maintains strong attribution performance even with passages retrieved directly.

D.2 Additional Baselines

Due to the differences in datasets and evaluation metrics, we include this baseline in the Appendix.

Learning to Plan and Generate Text with Citations (Fierro et al., 2024) conceptualizes plans as a sequence of questions that serve as blueprints for content generation and organization. Two variants of blueprint models are introduced: an abstractive model, where questions are generated from scratch, and an extractive model, where questions are copied from input passages.

For the comparative analysis of this method, we utilized the ASQA and ELI5 datasets, with top-5 reference passages retrieved directly via BM25 and GTR. The experimental results are shown in Table 7.

	ASQA		ELI5		Average	
	C	**A**	**C**	**A**	**C**	**A**
LongT5 3B (10-psg) +Blueprint$_A$ +Attribution	33.8	77.8	5.2	60.9	19.5	69.4
ReClaim w/IG (5-psg)	40.0	87.6	6.5	**76.1**	23.3	**81.9**
ReClaim w/IG (10-psg)	**40.8**	**88.5**	**8.0**	72.9	**24.4**	80.7

Table 7: Comparison with the baseline: Learning to Plan and Generate Text with Citations. **C** denotes accuracy, while **A** represents the attribution score, which is the F1 score of CAS and CRS.

The experimental results indicate that our approach outperforms this method in terms of both answer accuracy and attribution scores.

D.3 Faithfulness Results

The experimental results of the faithfulness metric for the ASQA and EXPERTQA datasets are shown in Figure 7 and Figure 8.

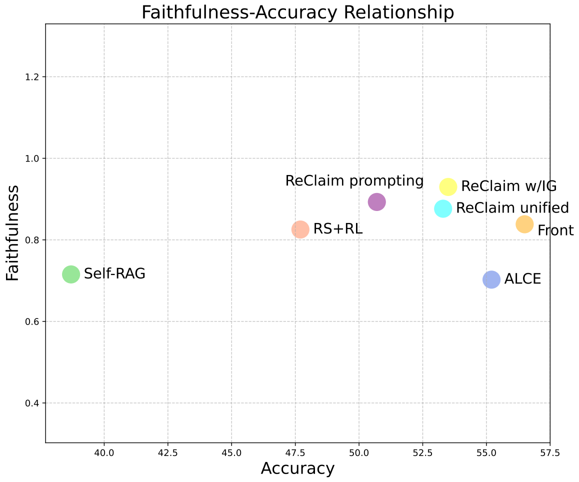


Figure 7: The faithfulness analysis results of the ASQA dataset.

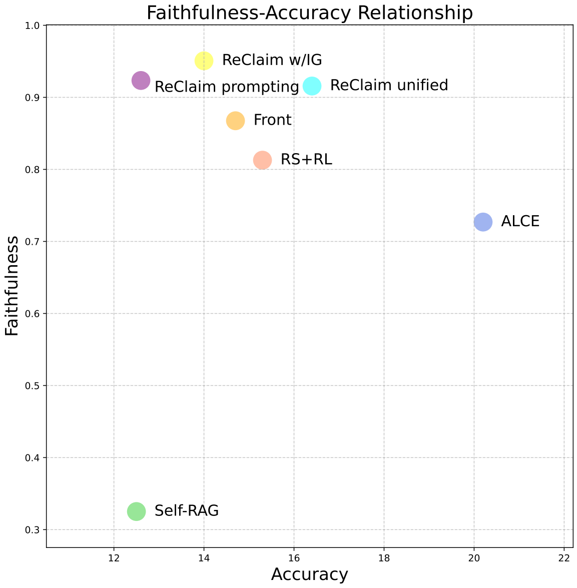


Figure 8: The faithfulness analysis results of the EXPERTQA dataset.

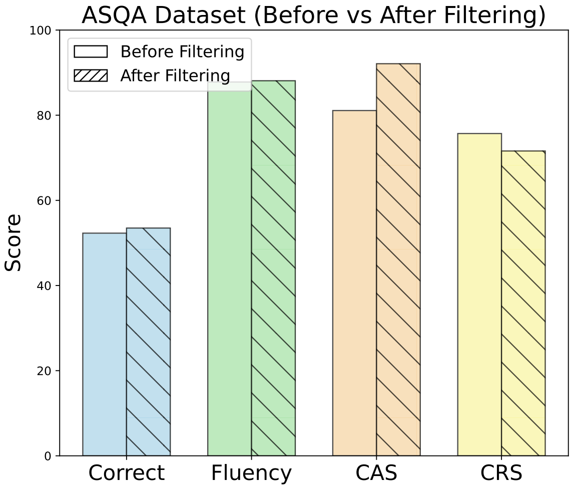


Figure 9: The comparative experimental results of the ASQA dataset.

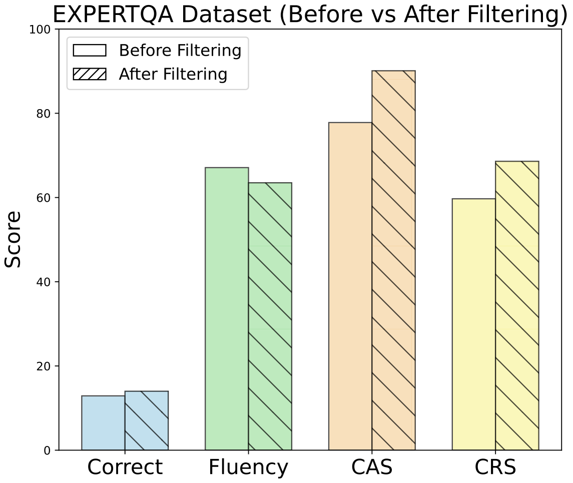


Figure 10: The comparative experimental results of the EXPERTQA dataset.

Question	Why does a dryer sheet help with my staticy shirt?
References	[1] Dryer sheets are specifically designed to help reduce static cling in clothes by neutralizing the electric charges that build up during the drying process. Just toss a sheet in with your clothes, and they should come out nice and static-free when they're done drying. [2] So, if you if you want your clothes to be ready to wear, but they're always covered in pet hair or are super clingy, then you definitely need to prevent static cling! Don't worry, we've all been there. Bounce Dryer Sheets will help your clothes lead a no-strings-attached lifestyle in no time. Just toss a sheet into the dryer with your clothes, and leave that static cling behind! [3] However, this coating can stick to your dryer and can be bad for the environment as well. Those looking to know how to reduce static on clothes without dryer sheets can find some solid, eco-friendly dryer sheet alternatives in many different places. [4] The most common way people know how to prevent dryer static on clothes is with dryer sheets. Dryer sheets are sheets that are coated in a fabric softener full of positively charged electrons. These bond to the negatively charged ones and keep static from happening. [5] Dryer static on clothes is one of the most annoying parts of doing laundry. It might seem like an unavoidable problem, but we have good news! There are some simple steps you can take to keep your laundry from becoming overly clingy or giving you static shocks.
Answer	According to the citation: <reference> The most common way people know how to prevent dryer static on clothes is with dryer sheets. Dryer sheets are sheets that are coated in a fabric softener full of positively charged electrons. These bond to the negatively charged ones and keep static from happening. </reference> We can know that: <claim> Dryer sheets are coated in a fabric softener full of positively charged electrons, which bond to the negatively charged ones in your clothes and keep static from happening. </claim>

Table 8: Example of the training dataset.

Dataset	**Samples**	**Question Type**	**#passages**	**Average Length**		
				Question	**Passages**	**Answer**
ASQA	948	Factoid (ambiguous)	Wikipedia (21M)	9.0	517.5	71.8
ELi5	1000	Why/How/What	Sphere (899M)	16.5	546.3	121.5
EXPERTQA	1000	ambiguous/unambiguous	Sphere (899M)	19.4	600.0	152.2

Table 9: Statistics of the test datasets.

Input	According to the citation: <reference> The most common way people know how to prevent dryer static on clothes is with dryer sheets. Dryer sheets are sheets that are coated in a fabric softener full of positively charged electrons. These bond to the negatively charged ones and keep static from happening. </reference>
Output	We can know that: <claim> Dryer sheets are coated in a fabric softener full of positively charged electrons, which bond to the negatively charged ones in your clothes and keep static from happening. </claim>

Table 10: Example of the training dataset for claim generation.

Question: When data is compressed or zipped, what is actually happening to the data?

Reference Passages:
[1] Title: Lossless vs. Lossy Compression: What's the Difference?
Text: data in several different ways, balancing fidelity and efficiency for functional and presentable data on the egress end. A common implementation of lossless file-compression includes the use of Huffman coding, whose redundancy-limiting algorithm recognizes patterns in groups in order to conserve time, space and other resources. The model is able to compress and decompress digital media such that the output perfectly matches the input. The Zip file archival tool is a well-known format that supports lossless compression. Zip files compress digital media into much smaller data (often given the 2̌018.zip2̌019 file extension) that can be uncompressed into their original form,
[2] Title: What is MP3? (Designing Web Audio)
Text: space, try encoding in mono. To make sure you trap all possible spatial data, encode in stereo mode. Most users find that joint stereo is adequate for most purposes. As mentioned earlier, compressing a WAV file with zip doesn't shave much off the file size, which is why psychoacoustics are employed. However, the MP3 encoding process actually does employ the classic Huffman encoding algorithm. After all psychoacoustic methods have been applied, the Huffman encoding pass seeks out and compresses any remaining redundancies in the bit pattern. It's as though zip-type encoding were being run internally on the psychoacoustically encoded data
[3] Title: US Treasury: addendum | Zip (File Format) | Comma Separated Values
Text: most large files on the Internet are compressed. File compression reduces the size of a file and the time it takes to download. Compression software uses complex mathematical equations to scan a file for repeating patterns in the data. It replaces the data with smaller codes that take up less room. This can dramatically decrease the size of report files. Viewing a 2̌01cZipped2̌01d file To view compressed files, you need a compatible decompression program that can read these codes and convert the data back to its original form. To decompress a .zip file you need a utility like WinZIP2̌122 (an
[4] Title: An introduction to data compression
Text: data and try to find redundant parts of that data and remove those redundancies. That wasn't too simple, but the definition needs to be somewhat complex in order to differentiate compression from some other technologies. 2. How is compression accomplished? Can you give us a picture of what actually happens to the data? Basically, they're trying to find common pieces of data blocks that they can get rid of, shrink, remove or substitute with smaller patterns. The more of those things they can find, the more it can compress. A perfect example to illustrate how that works is if you
[5] Title: Introduction to Data Compression - Data Compression | Coursera
Text: we get into the topic. so, just specific applications that, maybe are familiar for file compression. All file systems and, and disks have built-in, compression technologies. Such as, as zip, and b-zip and many others of similar type. Technologies. And the multimedia, everybody's familiar with, the JPEG and MP3 and MPEG, and all those sorts of things for images, sound and video. Those are all about data compression. and for communication, now, that's, what has, what enabled, fax, and also enables new technologies, like Skype, the ability to, reduce the, amount of data that you actually need to send. for
(Irrelevant passage)

Gold Answer: For most algorithms, the compressor will search for a pattern that appears multiple times within the data, and then replace that pattern with a shorter symbol, and create an entry in a "dictionary" recording what each symbol means. Compressing algorithms can get a little more complex than that, but that's the general idea.

Model Answer: According to the citation: <reference> Compression software uses complex mathematical equations to scan a file for repeating patterns in the data. It replaces the data with smaller codes that take up less room. </reference> We can know that: <claim> Compression software uses complex mathematical equations to scan a file for repeating patterns in the data and replaces the data with smaller codes that take up less room. </claim> According to the citation: <reference> A common implementation of lossless file-compression includes the use of Huffman coding, whose redundancy-limiting algorithm recognizes patterns in groups in order to conserve time, space and other resources. </reference> We can know that: <claim> A common implementation of lossless file-compression includes the use of Huffman coding, whose redundancy-limiting algorithm recognizes patterns in groups in order to conserve time, space and other resources. </claim> According to the citation: <reference> After all psychoacoustic methods have been applied, the Huffman encoding pass seeks out and compresses any remaining redundancies in the bit pattern. </reference> We can know that: <claim> After all psychoacoustic methods have been applied, the Huffman encoding pass seeks out and compresses any remaining redundancies in the bit pattern. </claim> According to the citation: <reference> Basically, they're trying to find common pieces of data blocks that they can get rid of, shrink, remove or substitute with smaller patterns. The more of those things they can find, the more it can compress. </reference> We can know that: <claim> The more of these common pieces of data blocks that can be found, the more it can compress. </claim>

Table 11: Example of the test data.

Step	Model	Input	Output
1	ReferModel	Instruction: Given the Question and References below, provide an answer for the Question that is generated using information exclusively from the References(some may be irrelevant). ... # Question: Who has the highest goals in world football? # References: "Football Player of the Century", by IFFHS International Federation of Football History and Statistics, 1999, "South America Football Player of the Century", by IFFHS International Federation of Football History and Statistics. Pelé's 1281 goals are recognized by FIFA as the highest total achieved by a professional footballer, although the Soccer Statistic Foundation (rssf) recognizes only 767 goals in official mode, occupying the third place after Josef Bican (805) and Romario (772). ... Output:	According to the citation: <reference> Pelé's 1281 goals are recognized by FIFA as the highest total achieved by a professional footballer, although the Soccer Statistic Foundation (rssf) recognizes only 767 goals in official mode, occupying the third place after Josef Bican (805) and Romario (772). </reference>
2	ClaimModel	According to the citation: <reference> Pelé's 1281 goals are recognized by FIFA as the highest total achieved by a professional footballer, although the Soccer Statistic Foundation (rssf) recognizes only 767 goals in official mode, occupying the third place after Josef Bican (805) and Romario (772). </reference>	We can know that: <claim> Pelé's 1281 goals are recognized by FIFA as the highest total achieved by a professional footballer, although the Soccer Statistic Foundation (rssf) recognizes only 767 goals in official mode, occupying the third place after Josef Bican (805) and Romario (772). </claim>
3	ReferModel	Instruction: Given the Question and References below, provide an answer for the Question that is generated using information exclusively from the References(some may be irrelevant). ... # Question: Who has the highest goals in world football? # References: "Football Player of the Century", ... Output: According to the citation: <reference> Pelé's 1281 goals are recognized by FIFA ... </reference> We can know that: <claim> Pelé's 1281 goals are recognized by FIFA ... </claim>	According to the citation: <reference> Rec.Sport.Soccer Statistics Foundation (RSSSF) estimates that he scored at least 805 goals in all competitive matches, which would make him the most prolific scorer of all time. </reference>
4	ClaimModel	According to the citation: <reference> Pelé's 1281 goals are recognized by FIFA ... </reference> We can know that: <claim> Pelé's 1281 goals are recognized by FIFA ... </claim> According to the citation: <reference> Rec.Sport.Soccer Statistics Foundation (RSSSF) estimates that he scored at least 805 goals in all competitive matches, which would make him the most prolific scorer of all time. </reference>	We can know that: <claim> Rec.Sport.Soccer Statistics Foundation (RSSSF) estimates that he scored at least 805 goals in all competitive matches, which would make him the most prolific scorer of all time. </claim>
...	...	...	...

Table 12: Example of the interleaving generation.