CiteLab: Developing and Diagnosing LLM Citation Generation Workflows via Human-LLM Interaction

Jiajun Shen[1,3*], **Tong Zhou**[1*], **Yubo Chen**[1,2†], **Kang Liu**[1,2,4], **Jun Zhao**[1,2]
[1]The Key Laboratory of Cognition and Decision Intelligence for Complex Systems
Institute of Automation, Chinese Academy of Sciences
[2]School of Artificial Intelligence, University of Chinese Academy of Sciences
[3]University of Chinese Academy of Sciences
[4]Shanghai Artificial Intelligence Laboratory
shenjiajun21@mails.ucas.ac.cn, tong.zhou@ia.ac.cn
{yubo.chen, kliu, jzhao}@nlpr.ia.ac.cn

Abstract

The emerging paradigm of enabling Large Language Models (LLMs) to generate citations in Question-Answering (QA) tasks is lacking in a unified framework to standardize and fairly compare different citation generation methods, leading to difficulties in reproduction and innovation. Therefore, we introduce Citeflow, an open-source and modular framework fostering reproduction and the implementation of new designs. Citeflow is highly extensible, allowing users to utilize four main modules and 14 components to construct a pipeline, evaluate an existing method, and understand the attributing LLM-generated contents. The framework is also paired with a visual interface, Citefix, facilitating case study and modification of existing citation generation methods. Users can use this interface to conduct LLM-powered case studies according to different scenarios. Citeflow and Citefix are highly integrated into the toolkit `CiteLab`, and we use an authentic process of multiple rounds of improvement through the Human-LLM interaction interface to demonstrate the efficiency of our toolkit on implementing and modifying citation generation pipelines. `CiteLab` is released at https://github.com/SjJ1017/CiteLab.

1 Introduction

Large Language Models (LLMs) (OpenAI, 2024; AI@Meta, 2024) possess the ability to store world knowledge (Du et al., 2024; Jin et al., 2024b; Chenhao Wang, 2025) and can handle multiple NLP tasks like translation (Yang Zhao, 2023). They demonstrate especially strong performance on Question Answering (QA) (Kamalloo et al., 2023) on different scenarios such as Commonsense QA (Talmor et al., 2019), long-form QA (Stelmakh et al., 2023; Min et al., 2020) and Multi-hop QA (Ho et al., 2020; Yang et al., 2018), but they can still inevitably produce hallucinated responses that are non-factual (Huang et al., 2023), nonsensical or irrelevant to the input (Xu et al., 2024c), reflecting the ongoing challenges in ensuring factual accuracy. Given the challenges above, Retrieval Augmented Generation (RAG) (Lewis et al., 2021) and citation generation (Gao et al., 2024) serve as an efficient way to make the answers of models accurate, more verifiable, and explainable.

Given the urgent need, ALCE (Gao et al., 2023b) developed basic methods to enable LLMs to generate citations in QA tasks and propose metrics for evaluating the quality of citations. Following ALCE's contribution, there are other methods that either use training (Huang et al., 2024a; Li et al., 2024a; Ye et al., 2024b; Huang et al., 2024b) or construct complicated pipelines to enhance the ability of generative models in citing external documents (Zhang et al., 2024a; Sun et al., 2024; Lee et al., 2023; Fierro et al., 2024; Qian et al., 2024). Another category related to citation generation is LLM attribution (Jain et al., 2023; Xu et al., 2024b; Gao et al., 2023a; Sun et al., 2023; Huang et al., 2024c; Cattan et al., 2024; Abolghasemi et al., 2024), which refers to the capacity of an LLM to generate and provide evidence (Li et al., 2023).

Despite considerable recent progress, there are still problems with regard to two main aspects.

Reproducibility and flexibility on citation generation tasks. Different works are distinguished largely by their implementation, hence the difficulty in reproducing. Low reproducibility not only increases deployment costs but also leads to the problem of comprehensive and fair horizontal comparisons between different methods. The lack of flexibility of different methods makes it difficult to integrate and improve various design concepts, thereby reducing the adaptability of the approach

[*]These authors contributed equally to this work.
[†]Corresponding author.
A demonstration video for CiteLab is available at https://youtu.be/aWuIG2OY7e8.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations), pages 490–501
July 27 - August 1, 2025

System	Custom Workflows	Citation Evaluation	Case Analysis	Live Testing	Workflow Modification
Langchain (Chase, 2022)	✓	✓	×	×	×
FalshRAG (Jin et al., 2024a)	✓	×	×	✓	×
RAGViz (Wang et al., 2024)	×	×	✓	✓	×
Low-code LLM (Cai et al., 2024)	✓	×	×	✓	✓
RAGLAB (Zhang et al., 2024b)	✓	×	×	×	×
AGREE (Ye et al., 2024a)	×	✓	✓	×	×
CiteLab	✓	✓	✓	✓	✓

Table 1: Comparison between CiteLab and other toolkits. Post-hoc analysis allows users to perform diagnostics and interpret the results. Live testing and workflow modification refer to the capability of conducting custom tests and modifying the workflow via the interface.

to different datasets and scenarios. The lack of flexibility of different methods makes it difficult to integrate and improve various design concepts, thereby reducing the adaptability of the approach to different datasets and scenarios. The lack of flexibility of different methods makes it difficult to integrate and improve various design concepts, thereby reducing the adaptability of the methods to different datasets and scenarios.

Lack of an interactive interface for efficient diagnosing and improving the citation workflow. Interactive visualization can significantly reduce the difficulty of use and facilitate case analysis. Though previous works developed a number of useful open-source toolkits or systems (Table 1) with user-friendly visualizations or RAG, these works cannot fully resolve the issue of time-consuming and labor-intensive workflow diagnosis, making them difficult to optimize citation-based methods. Due to the lack of integration with workflow frameworks, some attribution visualization works are only convenient for qualitative analysis and are difficult to use for improvement and innovation.

Given the problems above, a toolkit that integrates different design concepts flexibly and offers an easy-to-use interface is crucial for fast workflow implementation, diagnosis, and innovation. Therefore, we present CiteLab, an open-source, extensible, and user-friendly toolkit to facilitate research on the LLM citation generation task.

CiteLab offers a specially designed framework, Citeflow, for implementing citation generation workflows, containing four different types of modules: **Input**, **Generator**, **Enhancing Module**, and **Evaluator**, which are combined in a pipeline. The extensible modules and their flexible interconnection satisfy various needs of different implementations and comprehensive evaluation. This framework handles the problem of low reproducibility and insufficiency of flexibility of the existing methods. CiteLab also includes Citefix, a visual interactive interface highly compatible with Citeflow, enabling users to browse workflows, data, and interpretable post-hoc attributions of their own citation generation design and results. Additionally, it allows low-code utilization of large models for method summarization, case analysis, targeted workflow modification, and custom testing. The interface contains an AI-powered assistant, which efficiently analyzes the selected cases and gives helpful feedback and advice on improving the workflow. We validated the effectiveness of human-LLM collaboration in improving citation generation workflows through multi-round interactions. Our contributions can be summarized as follows:

- We propose a framework, Citeflow, which modularizes citation tasks containing 14 components and 16 functions derived by abstracting the ideas of existing methods, improving the reproducibility and evaluation in comprehensiveness of citation.
- We design a toolkit, CiteLab, which integrates the framework with a compatible visual interactive interface, Citefix, through which users can easily perform case studies efficiently and modify the workflow for optimization.
- We demonstrate convenient reproduction and effective diagnosis through a practical example, which indicates that it can facilitate the research and application of citation. Through human-LLM collaboration to improve workflows, we achieve a new state-of-the-art method, self-planning-RAG.

2 Related Work

2.1 Retrieval Augmented Generation

As LLMs can still inevitably produce hallucinated responses, Retrieval Augmented Generation (RAG) is a method introduced by Lewis et al. (2021) that improves text generation by retrieving external knowledge and generating. This approach helps generate more accurate and up-to-date answers, making it useful for tasks like question answering and creating content.

2.2 LLM Citation Generation

ALCE (Gao et al., 2023b) is the first attempt systematically to develop some basic methods to enable LLMs to generate citations in QA tasks and propose metrics for evaluating the quality of citations, showing there is still room for improvement concerning citation generation. Following ALCE's contribution, there are other methods that use training or construct complicated pipelines to enhance the ability of generative models to cite external documents. For example, Fierro et al. (2024) found the black-box generation is not factually faithful, so they use blueprint models to generate plans or blueprints and the output can be traced back to the blueprint to generate an explicit in-line citation. Verifiable Text Generation (VTG) (Sun et al., 2024) uses verifiers, evidence finder, retriever in case the documents do not support the output, and a simplifier to simplify citations to improve citation quality.

2.3 LLM Attribution

Another category related to citation generation is LLM attribution, which refers to the capacity of an LLM to generate and provide evidence (Li et al., 2023). For instance, Recitation Augmented Language Models(Sun et al., 2023), learns to sample documents from LLM's self-knowledge and construct a path of attributing passages to generate the final answer, although this task will not generate a citation, tracing back to the document that LLM refers to is possible, and a proper citation can visualize how LLM attribute from given documents or self-knowledge.

3 Features

3.1 Modular Citation Generation Framework

In this section, we will introduce the design of `CiteLab`, the details of different modules, and how they can form an integrated working pipeline of citation generation. We show our design in Figure 1.

INPUT handles data loading and prompt creation, managing user queries and the document corpus.

GENERATORcontains the LLM responsible for generating answers and citations, supporting various models (including GPT models, Llama, and others) and generation strategies (direct or iterative) A **GENERATOR**supports different frameworks, including huggingface, vllm, fastchat, and APIs like openai API to implement the generation according to the need.

ENHANCING MODULE. To summarize the modules used by different designs and improve reusability, we classify the functional modules into four categories: retriever, planner, assessor, and editor, as shown in Table 2. They can be used individually or collaboratively, providing sufficient flexibility for the construction of a citation generation pipeline. **ENHANCING MODULE** can be categorized into four types according to the functionality: (1) A retriever performs retrieval during the generation process. It can not only retrieve knowledge by *relevance*, like using bm-25 or dense passage retrieval (Karpukhin et al., 2020) but also get documents in the data store by a *summary* or samples documents from LLMs or even the **GENERATOR** itself (*inner*). (2) A Planner will process the query and documents in advance to help LLM generate responses. (3) A Feedbacker can automatically evaluate the draft answer in the process to guide the modules to generate a better response. (4) An Output editor can modify the response after generation to improve the citation quality or answer quality.

Methods	Feedbacker	Retriever	Planner	Editor
VANILLA				
Rerank	reranker			
INTERACT*		summary		
AnG*			attributer	
Bluprint			blueprint	
AAR	scorer			reviser
Citation Augmented		relevance		
VTG	verifier			simplifier
recitation		inner		
self-RAG*	reranker	relevance		

Table 2: The usage of different modules and ways of generation in different methods. Methods marked with (*) use iterative **GENERATOR**while others use direct ones.

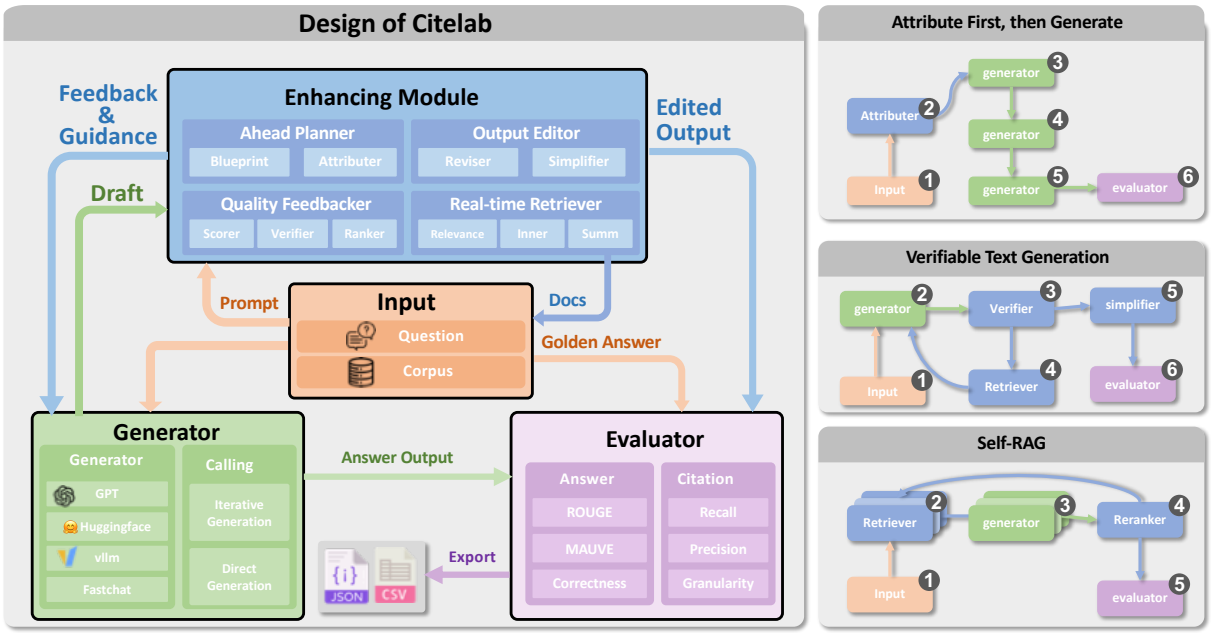


Figure 1: The modular design of CiteLab. On the left, we show four main modules in CiteLab and how they interact with other modules, as well as some predefined components and their abilities; on the right, we illustrate three baseline implementations in our framework and show the data flow during the running of their pipelines.

EVALUATOR is a module that evaluates or scores the output. There are some predefined metrics that can be set easily, such as ROUGE for answer quality and MAUVE for fluency, citation precision and recall in ALCE benchmark, a citation precision and recall metric with granularity (Zhao et al., 2024) for citation quality, and dataset-specific metrics for answer correctness (e.g., STR-EM for ASQA, claims for ELI5). Manually defined other metrics are also possible.

The modular design allows researchers to mix and match components, creating new citation-generation recipes by combining different modules, facilitating both the reproduction of existing methods and the exploration of new approaches.

3.2 Integrated Post-hoc Attribution with different Granularity

To help further conduct case studies and analysis on citation generation results, we integrated post-hoc attribution methods into our framework. The final answer will be automatically attributed to the documents that the answer refers to. Given the post-hoc attribution scores, users can qualitatively assess the answer and the citation generation process. We integrate three attribution methods with different granularity: document-level, span-level, and token-level.

Document-level Attribution. Given an LLM-generated answer and a set of documents, the attribution score of a document d_i is defined as the increase in the perplexity of the answer when d_i is removed from the text. A higher attribution score indicates that the document plays a more critical role in supporting the generated answer.

Span-level Attribution. We use CONTEXTCITE (Cohen-Wang et al., 2024) for span-level attribution. CONTEXTCITE uses a surrogate model to track the information sources of LLM-generated content. This method splits the knowledge source by sentence boundaries and returns attributing scores for each sentence.

Token-level Attribution. We use MIRAGE (Qi et al., 2024), an internals-based answer attribution method that identifies context-sensitive tokens and calculates their attributing scores to each token in the context.

3.3 Visualizations for Case Analysis

To facilitate further case analysis, help users to summarize cases and make real-time modifications and tests on the pipeline, we visualized our pipeline, data stream and evaluation results with attribution scores.

3.3.1 Interactive Visualizations

In order to optimize reference generation for different scenarios, we adapted a visualization analysis tool for our framework as in Figure 2. The visu-

alization of our framework illustrates the entire workflow and detailed configurations, with the data stream of the workflow in the corresponding panel. For each data point, the corresponding information and the result, including retrieved documents, will be presented. As post-hoc attribution methods are integrated into our framework, the attribution score distribution for each output sentence with different granularity is also displayed. Users can define their custom data and run the workflow via the interface to quickly test the effectiveness of the pipeline. Our visualization is designed especially for our framework, making it compatible with the design of various citation generation pipelines.

3.3.2 Diagnostic Tool for Case Analysis

Despite the interface, case analysis is still a time-consuming and labor-intensive task for humans, as they need to inductively analyze a large amount of test data and identify problems. We recognize the good alignment between large models and human preferences (Liu et al., 2024), as well as the information extraction (Xu et al., 2024a) and inductive capabilities of long-context models (Bowen et al., 2024; Lee et al., 2025) to solve a wide range of real-world problems (Azaria et al., 2024; Niu et al., 2025). Therefore, we have integrated a diagnostic tool based on Human-LLM interaction to improve the efficiency of case analysis.

Inductive summarization. The visual interface allows users to use an LLM assistant to summarize case issues. Users can easily identify failure cases through the interface, and by simply selecting certain data points, the LLM assistant will automatically read data and provide feedback on the common patterns of failure cases and categorize the summaries.

Case analysis and advice generation. Users can use the interface to analyze the causes of a specific issue. The assistant can read the selected case and the workflow automatically to provide modification suggestions based on the design of the pipeline. Given the suggestions, users are able to modify the pipeline through an interactive interface while conducting customized data tests.

4 Use Case

In this section, we showcase how to utilize our framework to easily evaluate citation generation methods, find insightful results through comparison, conduct a case study, and improve the existing methods via our interactive interface.

4.1 Baselines

We evaluate 11 baselines in total using the state-of-the-art open-source and closed-source LLMs, GPT-4o (OpenAI, 2024) and Llama3-8B-Instruct (AI@Meta, 2024) on ASQA dataset. Three sorts of baselines are included: (1) **ALCE baselines.** ALCE-VANILLA, SNIPPET, SUMM, ALCE INTERACT (Gao et al., 2023b). (2) **Citation based methods.** AAR (Lee et al., 2024), VTG (Sun et al., 2024) , Citation Enhanced (Li et al., 2024b), , Attribute First, then Generate (Slobodkin et al., 2024) and Blueprint (Fierro et al., 2024). (3) **RAG or Attribution-based.** Recitation Augmented (Sun et al., 2023) and self-RAG (Asai et al., 2023). Detailed implementation and settings are shown in Appendix A.

4.2 Results

We use metrics from ALCE for evaluation, including fluency, correctness, rouge, citation recall, and precision, as well as citation granularity. We show the full results and our analysis in Appendix B.

4.3 Multiple Rounds of Improvement

After the evaluation on different baselines, we find self-RAG achieves a decent performance on ASQA dataset. However, the existing failed cases indicate that this method can still be further improved. We demonstrate how our toolkit effectively facilitates modification and innovation on an implemented pipeline through multiple rounds of interaction between humans and LLMs. We evaluate the method on the ASQA dataset with Llama3-8B-Instruct after each step and show the improvement of the performance in Figure 3.

4.3.1 Round 1, Revision on Prompt

We selected dozens of failed cases with low citation quality and automatically provided them to the LLM through our interactive interface for summarization and improvement suggestions. Our assistant has identified a frequently occurring issue where the correct answer contains multiple entities, but the retrieved articles cover only one entity or even retrieve the same article repeatedly. As a result, the generated output consists of several sentences repeating the same fact, lacking diversity and reducing the overall coverage of the correct answer. Therefore, the assistant suggests modifying the prompt for the query generator to enhance query diversity and provide some suitable alternatives. We update the template for the input of the

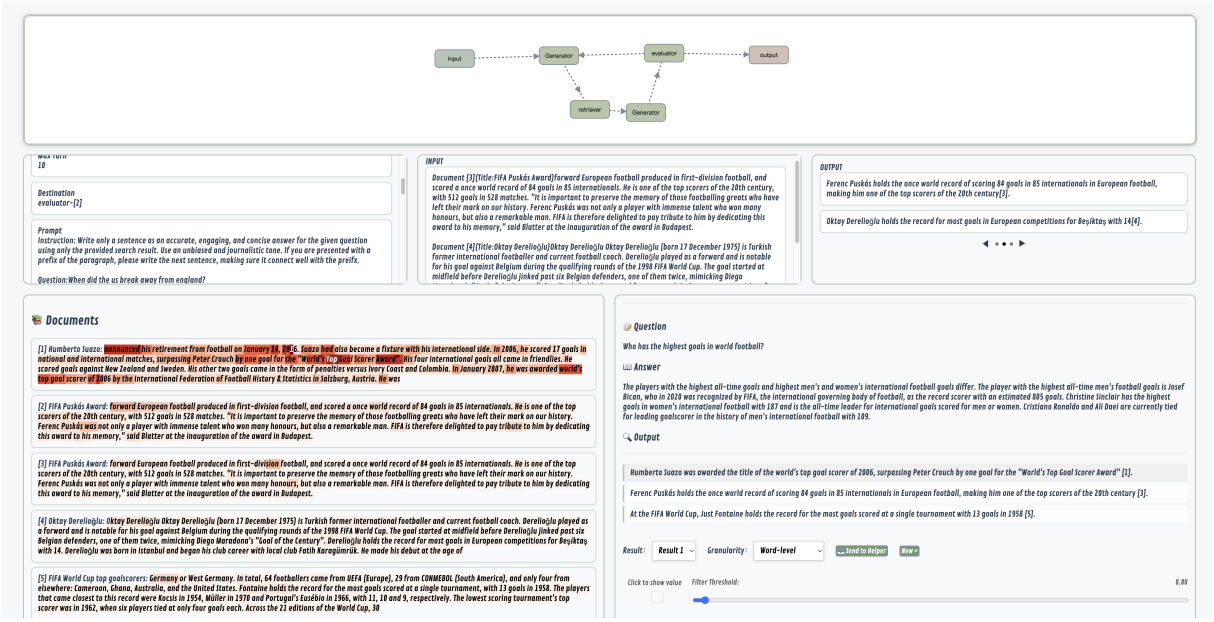


Figure 2: Visual Interface of Citefix. The panel at the top shows the pipeline, the panel in the middle presents configurations and data stream of the selected module, and the panel at the bottom shows the results.

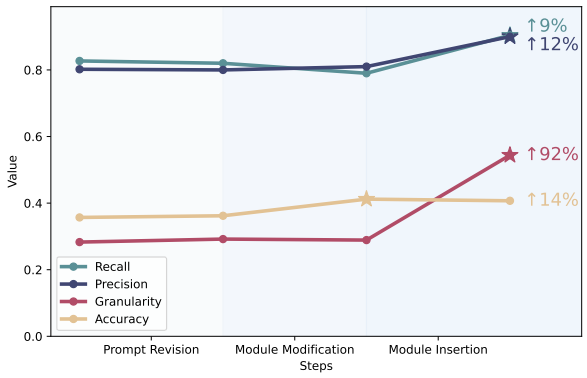


Figure 3: Performance after each modification.

query generator of the workflow. Figure 4 demonstrates the revision.

```
Original: Please generate a natural language
query to help find relevant documents.

Modified: Please generate a natural language
query to help find relevant documents. If
previous queries are provided, you should
focus on an alternative perspective or
subtopic different from the provided ones,
enhancing diversity in retrieved documents.
```

Figure 4: Revision on prompt in the first round

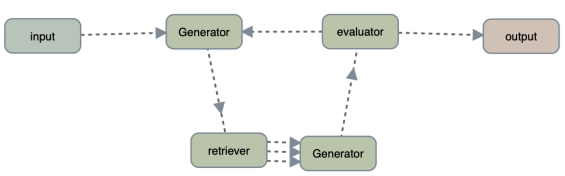
4.3.2 Round 2, Module Modification

After evaluating the modified workflow again, we observe a slight increase in average answer accuracy. To validate the effectiveness of the modification, we ask the assistant to analyze whether the previous problem has been addressed. We select certain data with low answer accuracy based on the evaluation results without checking each piece of data manually, and send it to the assistant. Unfortunately, there still exists a number of answers with low diversity. The assistant points out that the potential problem is the iterative generating process, in which a new generated sentence will follow the previous sentence, and this results in the generated answer potentially being unfaithful to the documents and reduces the diversity of the answer.

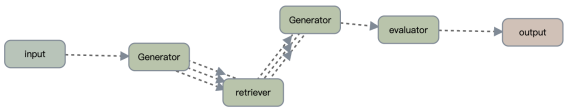
Following the advice, we modify the query generator to allow it generate multiple diverse queries as a list, and the process after the retriever will automatically switch to parallel, given a list of inputs.

4.3.3 Round 3, Insertion of New Modules

We witness a considerable improvement in answer quality after the second round of modifications. However, the citation quality still needs to be improved. The assistant analyzes some results with low citation recall and finds a serious issue: if the retrieved documents are not relevant to the question, the workflow still forces the generator to output an answer sentence and automatically adds a citation. As a consequence, the answer includes redundant citation, even if the output is not generated from the retrieved document, but an improvised or refusal answer. The assistant also notices that



(a)



(b)

Figure 5: Workflow (a) before and (b) after modification

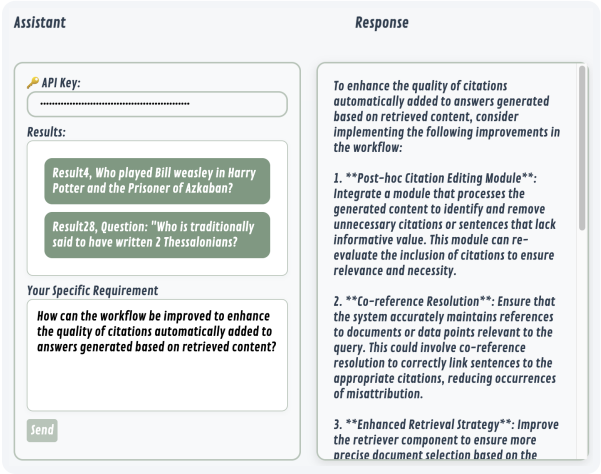


Figure 6: The suggestion generated by the assistant.

the granularity could be improved by an extraction module before the generator, as presented in Figure 6. Given the existing problems, the advice is to insert an extraction module and a citation simplifier before and after the generator, respectively.

We apply the modification, and the results shows that the the final workflow, named self-planning-RAG, achieves a new state-of-the-art performance on both the quality of answer and citation.

5 Conclusion

To unify various methods for LLM citation generation and facilitate the exploration of citation generation tasks, we propose a user-friendly and extensible toolkit with a visual interface, `CiteLab`. We also present a use case to demonstrate the application, showing the usability and versatility of our framework. We conducted experiments on 11 baselines and, based on the best-performing one, applied `CiteLab` for multiple rounds of improvement and achieved SOTA results, demonstrating the efficiency of `CiteLab` in citation generation research.

6 Acknowledgment

This work is supported by the National Natural Science Foundation of China (No.U24A20335, No. 62176257). This work is also supported by Beijing Natural Science Foundation (L243006). This work is also supported by the Youth Innovation Promotion Association CAS.

7 Limitations

There are still areas for improvement in our evaluation. (1) We only conduct our experiment on two LLMs, GPT-4o and Llama3-8B-Instruct. The effectiveness of the toolkit can also be validated through more case studies, and the usage experiences and feedback reports from other users are also important for confirming its effectiveness. (2) The diagnostic process relies heavily on the assistant's interpretability since the assistant depends on an external LLM, and the user's understanding is also important in Human-LLM interaction. (3) The settings of the experiments could be improved, such as using the latest technologies to retrieve (Luo et al., 2024) and utilize documents.

References

Amin Abolghasemi, Leif Azzopardi, Seyyed Hadi Hashemi, Maarten de Rijke, and Suzan Verberne. 2024. Evaluation of attribution bias in retrieval-augmented large language models. *Preprint*, arXiv:2410.12380.

AI@Meta. 2024. Llama 3 model card.

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

Amos Azaria, Rina Azoulay, and Shulamit Reches. 2024. Chatgpt is a remarkable tool—for experts. *Data Intelligence*, 6(1):240–296.

Chen Bowen, Rune Sætre, and Yusuke Miyao. 2024. A comprehensive evaluation of inductive reasoning capabilities and problem solving in large language models. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 323–339, St. Julian's, Malta. Association for Computational Linguistics.

Yuzhe Cai, Shaoguang Mao, Wenshan Wu, Zehua Wang, Yaobo Liang, Tao Ge, Chenfei Wu, WangYou WangYou, Ting Song, Yan Xia, Nan Duan, and Furu Wei. 2024. Low-code LLM: Graphical user interface over large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics:*

Human Language Technologies (Volume 3: System Demonstrations), pages 12–25, Mexico City, Mexico. Association for Computational Linguistics.

Arie Cattan, Paul Roit, Shiyue Zhang, David Wan, Roee Aharoni, Idan Szpektor, Mohit Bansal, and Ido Dagan. 2024. Localizing factual inconsistencies in attributable text generation. *Preprint*, arXiv:2410.07473.

Harrison Chase. 2022. LangChain.

Yubo Chen Kang Liu Jun Zhao Chenhao Wang, Jiachun Li. 2025. A survey of recent advances in commonsense knowledge acquisition: Methods and resources. *Machine Intelligence Research*, 22:201–218.

Benjamin Cohen-Wang, Harshay Shah, Kristian Georgiev, and Aleksander Madry. 2024. Contextcite: Attributing model generation to context. *arXiv preprint arXiv:2409.00729*.

Pengfan Du, Sirui Liang, Baoli Zhang, Pengfei Cao, Yubo Chen, Kang Liu, and Jun Zhao. 2024. ZhuJiuknowledge: A fairer platform for evaluating multiple knowledge types in large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 3: System Demonstrations)*, pages 194–206, Mexico City, Mexico. Association for Computational Linguistics.

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.

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

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

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2024. Retrieval-augmented generation for large language models: A survey. *Preprint*, arXiv:2312.10997.

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

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, Seattle, United States. Association for Computational Linguistics.

Chengyu Huang, Zeqiu Wu, Yushi Hu, and Wenya Wang. 2024a. Training language models to generate text with citations via fine-grained rewards. *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. *Preprint*, arXiv:2408.04568.

Lei Huang, Xiaocheng Feng, Weitao Ma, Liang Zhao, Yuchun Fan, Weihong Zhong, Dongliang Xu, Qing Yang, Hongtao Liu, and Bing Qin. 2024c. Advancing large language model attribution through self-improving. *Preprint*, arXiv:2410.13298.

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

Palak Jain, Livio Baldini Soares, and Tom Kwiatkowski. 2023. 1-pager: One pass answer generation and evidence retrieval. *Preprint*, arXiv:2310.16568.

Jiajie Jin, Yutao Zhu, Xinyu Yang, Chenghao Zhang, and Zhicheng Dou. 2024a. Flashrag: A modular toolkit for efficient retrieval-augmented generation research. *CoRR*, abs/2405.13576.

Zhuoran Jin, Pengfei Cao, Yubo Chen, Kang Liu, Xiaojian Jiang, Jiexin Xu, Qiuxia Li, and Jun Zhao. 2024b. Tug-of-war between knowledge: Exploring and resolving knowledge conflicts in retrieval-augmented language models. *Preprint*, arXiv:2402.14409.

Ehsan Kamalloo, Nouha Dziri, Charles L. A. Clarke, and Davood Rafiei. 2023. Evaluating open-domain question answering in the era of large language models. *Preprint*, arXiv:2305.06984.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen tau Yih. 2020. Dense passage retrieval for open-domain question answering. *Preprint*, arXiv:2004.04906.

Dongyub Lee, Eunhwan Park, Hodong Lee, and Heuiseok Lim. 2024. Ask, assess, and refine: Rectifying factual consistency and hallucination in LLMs

with metric-guided feedback learning. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2422–2433, St. Julian's, Malta. Association for Computational Linguistics.

Dongyub Lee, Taesun Whang, Chanhee Lee, and Heuiseok Lim. 2023. Towards reliable and fluent large language models: Incorporating feedback learning loops in qa systems. *Preprint*, arXiv:2309.06384.

Taewhoo Lee, Chanwoong Yoon, Kyochul Jang, Donghyeon Lee, Minju Song, Hyunjae Kim, and Jaewoo Kang. 2025. Ethic: Evaluating large language models on long-context tasks with high information coverage. *Preprint*, arXiv:2410.16848.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2021. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Preprint*, arXiv:2005.11401.

Dongfang Li, Zetian Sun, Baotian Hu, Zhenyu Liu, Xinshuo Hu, Xuebo Liu, and Min Zhang. 2024a. Improving attributed text generation of large language models via preference learning. *Preprint*, arXiv:2403.18381.

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

Weitao Li, Junkai Li, Weizhi Ma, and Yang Liu. 2024b. Citation-enhanced generation for llm-based chatbots. *Preprint*, arXiv:2402.16063.

Wenhao Liu, Xiaohua Wang, Muling Wu, Tianlong Li, Changze Lv, Zixuan Ling, Jianhao Zhu, Cenyuan Zhang, Xiaoqing Zheng, and Xuanjing Huang. 2024. Aligning large language models with human preferences through representation engineering. *Preprint*, arXiv:2312.15997.

Man Luo, Xin Xu, Zhuyun Dai, Panupong Pasupat, Mehran Kazemi, Chitta Baral, Vaiva Imbrasaite, and Vincent Y Zhao. 2024. Dr.icl: Demonstration-retrieved in-context learning. *Data Intelligence*, 6(4):909–922.

Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2020. AmbigQA: Answering ambiguous open-domain questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5783–5797, Online. Association for Computational Linguistics.

Tong Niu, Haoyu Huang, Yu Du, Weihao Zhang, Luping Shi, and Rong Zhao. 2025. General automatic solution generation for social problems. *Machine Intelligence Research*, 22(1):145–159.

OpenAI. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Jirui Qi, Gabriele Sarti, Raquel Fernández, and Arianna Bisazza. 2024. Model internals-based answer attribution for trustworthy retrieval-augmented generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 6037–6053, Miami, Florida, USA. Association for Computational Linguistics.

Haosheng Qian, Yixing Fan, Ruqing Zhang, and Jiafeng Guo. 2024. On the capacity of citation generation by large language models. *Preprint*, arXiv:2410.11217.

Aviv Slobodkin, Eran Hirsch, Arie Cattan, Tal Schuster, and Ido Dagan. 2024. Attribute first, then generate: Locally-attributable grounded text generation. *Preprint*, arXiv:2403.17104.

Ivan Stelmakh, Yi Luan, Bhuwan Dhingra, and Ming-Wei Chang. 2023. Asqa: Factoid questions meet long-form answers. *Preprint*, arXiv:2204.06092.

Hao Sun, Hengyi Cai, Bo Wang, Yingyan Hou, Xiaochi Wei, Shuaiqiang Wang, Yan Zhang, and Dawei Yin. 2024. Towards verifiable text generation with evolving memory and self-reflection. *Preprint*, arXiv:2312.09075.

Zhiqing Sun, Xuezhi Wang, Yi Tay, Yiming Yang, and Denny Zhou. 2023. Recitation-augmented language models. *Preprint*, arXiv:2210.01296.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

Tevin Wang, Jingyuan He, and Chenyan Xiong. 2024. RAGViz: Diagnose and visualize retrieval-augmented generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 320–327, Miami, Florida, USA. Association for Computational Linguistics.

Derong Xu, Wei Chen, Wenjun Peng, Chao Zhang, Tong Xu, Xiangyu Zhao, Xian Wu, Yefeng Zheng, Yang Wang, and Enhong Chen. 2024a. Large language models for generative information extraction: A survey. *Preprint*, arXiv:2312.17617.

Shicheng Xu, Liang Pang, Huawei Shen, Xueqi Cheng, and Tat-Seng Chua. 2024b. Search-in-the-chain: Interactively enhancing large language models with search for knowledge-intensive tasks. *Preprint*, arXiv:2304.14732.

Ziwei Xu, Sanjay Jain, and Mohan Kankanhalli. 2024c. Hallucination is inevitable: An innate limitation of large language models. *Preprint*, arXiv:2401.11817.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *Preprint*, arXiv:1809.09600.

Chengqing Zong Yang Zhao, Jiajun Zhang. 2023. Transformer: A general framework from machine translation to others. *Machine Intelligence Research*, 20:514–538.

Xi Ye, Ruoxi Sun, Sercan Arik, and Tomas Pfister. 2024a. Effective large language model adaptation for improved grounding and citation 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 6237–6251, Mexico City, Mexico. Association for Computational Linguistics.

Xi Ye, Ruoxi Sun, Sercan Ö. Arik, and Tomas Pfister. 2024b. Effective large language model adaptation for improved grounding and citation generation. *Preprint*, arXiv:2311.09533.

Jingyu Zhang, Marc Marone, Tianjian Li, Benjamin Van Durme, and Daniel Khashabi. 2024a. Verifiable by design: Aligning language models to quote from pretraining data. *Preprint*, arXiv:2404.03862.

Xuanwang Zhang, Yunze Song, Yidong Wang, Shuyun Tang, et al. 2024b. RAGLAB: A modular and research-oriented unified framework for retrieval-augmented generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Association for Computational Linguistics.

Suifeng Zhao, Tong Zhou, Zhuoran Jin, Hongbang Yuan, Yubo Chen, Kang Liu, and Sujian Li. 2024. Awecita: Generating answer with appropriate and well-grained citations using llms. *Data Intelligence*, 6(4):1134–1157.

A Experiment Implementation and Settings

A.1 Baselines and metrics

We evaluate 11 baselines in total using the state-of-the-art open-source and closed-source LLMs, GPT-4o (OpenAI, 2024) and Llama3-8B-Instruct (AI@Meta, 2024) on ASQA dataset. **ALCE** VANILLA, SNIPPET, and SUMM directly prompt the LLM to generate citations using full documents, snippets, and summaries respectively. **ALCE** INTERACT (Gao et al., 2023b) uses document summaries and interactively provides full documents. **AAR** (Lee et al., 2024) asked the LLM to revise the answer, while **VTG** (Sun et al., 2024) will verify the answer and retrieve more supplementary documents for regeneration. **Citation Enhanced** (Li et al., 2024b) method retrieves documents after generation, and **Recitation Augmented** (Sun et al., 2023) sample documents from pre-training data. **Attribute First, then Generate** (Slobodkin et al., 2024) and **Blueprint** (Fierro et al., 2024) provides some attributing spans or questions to guide the generation. For **self-RAG** (Asai et al., 2023), we use our prompt version instead of a trained model to retrieve documents and generate sentence-by-sentence. We use metrics from ALCE for evaluation, including fluency, correctness, rouge, citation recall, and precision. We also evaluate the appropriate citation rate and the citation granularity.

A.2 Settings

We set max generated tokens to 500 to avoid too long answers and use \n as stop token. For Llama3-8B-Instruct, we use the model from huggingface and set the temperature to 0.5. and other configurations by default. For GPT-4o, we use the openai API. During our experiment, we used the same prompt for the two models.

For retrieving documents relevant to the query, we use 5 documents by default. However, for ALCE SUMM, ALCE SNIPPET, and ALCE INTERACT, we use 10 documents as they show the short summaries and snippets from the documents. Citation Augmented and self-RAG use real-time retrievers instead of a fixed number of document inputs, and we configured our retrievers to return the top-1 document at a time.

For evaluation of citation quality, we adopt a TRUE model (Honovich et al., 2022) to verify if the cited documents could entail the generated statement.

B Results

We show the full results on ASQA dataset in Table 3. We discuss the main results from the experiments below.

In our experiments, we find that a stronger base model improves citation quality and answer correctness, as seen in GPT-4o outperforming Llama3-8B-Instruct. Planning enhances answer accuracy, especially for powerful models like GPT-4o, while an editor significantly improves citation precision and recall, but enhancing citation granularity remains a challenge, as most models cite full documents. Methods like ALCE-SUMM and ALCE-SNIPPET attempt to cite summaries or snippets but risk correctness loss. Interestingly, Llama3-8B-Instruct shows better citation precision and recall when citing internal knowledge, despite reducing answer quality, suggesting further research potential.

C Implementation Details

In this section, we describe the implementation details for different baselines. For other baselines, we follow the original prompts and the structure they provided, but for Blueprint and self-RAG, we use In-Context-Learning (ICL) instead of a trained model to complete the sub-task in their design.

C.1 Blueprint Model

For the Blueprint Model, we use the abstractive model to produce general-purpose questions: the paragraph is the input and the question is the output. We use prompts to make LLMs generate questions. ALCE provides question-answer pairs for ASQA dataset, and in each pair the sub-question shows an aspect of answering the final question. We use these pairs to complete a 2-shot prompt for ICL. For answer generation, we adjust the ALCE prompt to make LLMs answer all the subquestions.

C.2 Prompt self-RAG

As for Llama3-8B and GPT-4o, there is no trained version for self-RAG, we use prompt to make the LLM retrieve documents and generate, then use an NLI model to evaluate if the document is supportive and the answer is useful, respectively in 3 segments. A reranker will find the best segment and the sentence is added to the answer. Similar to Attribute First, then Generate, We use generated sentences as prefixes to complement the sentence-by-sentence iterative generation.

		Fluency	Correct.	Citation					
	Model	(MAUVE)	(EM Rec.)	Rec.	Prec.	App.	Gran.	ROUGE-L	Length
ALCE VANILLA	llama3-8B	66.8	40.5	47.2	53.8	80.5	22.5	28.6	72.0
	GPT-4o	72.3	41.0	59.5	61.3	70.8	19.3	32.4	41.6
ALCE SUMM	llama3-8B	80.1	40.6	59.5	66.2	80.6	59.7	27.7	69.4
	GPT-4o	72.3	42.0	59.6	61.4	82.6	54.5	32.5	41.6
ALCE SNIPPET	llama3-8B	69.2	38.9	56.7	60.9	81.8	65.6	27.1	65.3
	GPT-4o	79.7	37.3	77.0	66.8	85.6	58.3	30.2	26.5
ALCE INTERACT	llama3-8B	68.0	30.3	30.6	56.1	84.1	17.2	21.5	56.6
	GPT-4o	72.6	39.9	41.2	45.0	72.0	12.0	30.4	67.3
Attribute, then Generate	llama3-8B	70.2	38.9	49.2	42.7	78.0	22.8	27.9	89.3
	GPT-4o	75.5	41.6	63.4	42.7	87.0	19.2	24.8	61.2
AAR	llama3-8B	69.4	38.9	37.8	47.8	74.1	28.1	27.0	122.8
	GPT-4o	72.2	46.0	52.4	58.7	77.8	20.9	31.5	59.0
Citation Enhanced	llama3-8B	59.2	31.0	30.9	40.8	54.0	27.2	24.8	48.7
	GPT-4o	65.3	41.3	49.8	52.8	55.3	27.0	29.6	40.6
VTG	llama3-8B	74.9	41.2	73.4	73.1	87.3	27.0	42.4	45.3
	GPT-4o	75.1	42.3	83.0	82.5	88.4	29.3	39.3	45.3
Blueprint	llama3-8B	70.0	40.8	68.5	71.3	87.5	22.5	31.2	75.8
	GPT-4o	78.2	41.2	68.5	83.0	83.6	19.8	27.2	75.8
Recitation Augmented	llama3-8B	61.2	33.6	47.6	55.0	62.5	14.4	34.5	129
	GPT-4o*	/	/	/	/	/	/	/	/
Self-RAG	llama3-8B	68.4	35.7	82.7	80.2	88.3	28.3	27.1	52.2
	GPT-4o	70.7	37.9	81.5	83.25	84.6	26.4	27.9	40.7
self-Planning -RAG(Ours)	llama3-8B	70.3	40.7	90.4	90.0	91.1	54.4	32.1	38.8

Table 3: ASQA results. *In recitation-augmented baseline, we only use Llama3-8B-Instruct because we found GPT-4o is too reluctant to recite verbatim documents in training data.