$\mathcal{A}^3$: Automatic Alignment Framework for Attributed Text Generation

Anonymous ACL submission

Abstract

Attributed text generation aims to enhance the reliability of content generated from large language models by providing citations for each claim, which thereby enables users to easily verify the correctness of the responses. However, the scarcity of high-quality training samples presents a significant challenge in aligning large language models to generate texts with citations, revealing considerable room for improvement in existing attribution systems. Besides, existing approaches of aligning large language models to follow user instructions can lead to an undue emphasis on irrelevant documents, which in turn reduces the quality of responses. To address the above problems, we propose **A**utomatic **A**lignment Framework for **A**ttributed Text Generation ($\mathcal{A}^3$), a novel framework designed to automatically generate high-quality attributed query-response pairs for both supervised fine-tuning and preference optimization stages without human annotation. With the help of $\mathcal{A}^3$, Mistral-7B can achieve a citation recall of **84.4** and a precision of **87.0** precision on ASQA, which notably surpasses GPT-4's citation recall of **73.0** and precision of **76.5**. [1]

1 Introduction

Recently, due to the convenience of natural language interaction, an increasing number of users prefer to employ Large Language Models (LLMs) for their information-seeking needs. However, despite the abundant knowledge obtained during pre-training, the outputs of LLMs can sometimes deviate from user instructions and contain hallucinations, significantly constraining their ability to satisfy information-seeking needs (Zhang et al., 2023b). Furthermore, due to the lack of clear attributions, it is also difficult to check the correctness of content generated from LLMs (Asai et al., 2024b). Therefore, to better satisfy the information-seeking needs, attributed text generation has recently gained significant attention from both academics and industry, which aims to enhance the reliability of generated content by providing citations for each claim (Li et al., 2023a).

Despite the significant importance of attributed text generation, existing open-source attribution systems exhibit considerable room for improvement. In the era of LLMs, constructing an attribution system typically involves a two-step process (Gao et al., 2023b; Malaviya et al., 2023). Firstly, an external retriever is used to get relevant passages. Subsequently, these passages, along with the user query, are incorporated into carefully designed templates. Finally, these templates serve as the input of LLMs and guide them to generate responses with proper citations. However, since existing LLMs are designed to follow user instructions, this approach can lead to models generating content based on irrelevant retrieved passages, thus undermining the quality of the attribution. Moreover, even if provided with relevant passages, it is still challenging to rely only on the task generalization capabilities of LLMs to follow user instructions and generate responses with correct citations that align with the question (Gao et al., 2023b). Overall, the lack of high-quality open-source data hinders the development of attributed text generation.

The challenge of obtaining high-quality open-source attributed text generation training data stems from several reasons. Firstly, the high cost of human annotation makes large-scale, human-annotated datasets unaffordable, which results in the necessity of automatic data generation. Besides, while existing commercial attribution systems, such as Bing Chat [2] and perplexity.ai [3], have achieved success, the open-source community can-

[1] Our code and dataset will be accessible at https://anonymous.

[2] https://www.bing.com/chat

[3] https://www.perplexity.ai/

not benefit from these commercial systems. Specifically, since there is no open access to the API of Bing Chat and it does not contain citations in the responses from the API of Perplexity.Ai, we can not use these commercial systems to generate attributed query-response as training data. Besides, due to the inability to obtain results of commercial systems on academic benchmarks, we can not directly compare our models with commercial systems like Bing Chat and Perplexity.Ai, which also hinders the development of this task. Finally, despite the strong general ability, the performance of existing LLMs (e.g., GPT-4) is far from satisfactory, which leads to poor results in directly using existing LLMs to generate attributed text generation data (Kamalloo et al., 2023).

To tackle the aforementioned challenges, we introduce a novel framework, the **A**utomatic **A**lignment Framework for **A**ttributed Text Generation ($\mathcal{A}^3$), which leverages underperforming attributed LLMs to automatically generate high-quality attributed text generation data without the need for human-annotated datasets. To achieve this goal, the $\mathcal{A}^3$ framework breaks down the difficult attributed text generation task into simple solvable ones, e.g., document clustering, text summarization, text entailment, etc. Specifically, the $\mathcal{A}^3$ framework comprises two main processes: data construction and model training. During the data construction process, we start by employing an entity-matching method to cluster multiple documents on relevant topics. Then, we utilize LLMs to construct Q&A pairs based on these interrelated documents. To reduce the effect of irrelevant documents, we introduce a noisy document augmentation strategy for constructing supervised fine-tuning data, alongside four strategies to generate dispreferred responses for the preference optimization stage. Finally, these constructed data are utilized in the model training process.

In conclusion, our work has the following contributions:

- We introduce the $\mathcal{A}^3$ framework, which can automatically high-quality training samples for attributed text generation without requiring human annotations;
- Leveraging $\mathcal{A}^3$, the Mistral-7B model achieves a citation recall of **84.4** and precision of **87.0** on ASQA, markedly outperforming GPT-4, which achieves a citation recall of **73.0** and precision of **76.5**;
- Our experimental results also demonstrate the effectiveness of our proposed framework in reducing the effects of noisy documents and avoiding irrelevant citations.

2 Automatic Alignment Framework for Attributed Text Generation

2.1 Attribute Text Generation Task Setup

Given the query Q_i and the retrieved passages $P_i = \{p_1, p_2, \ldots, p_n\}$, attributed text generation aims to generate a response R_i with citations $C_i = \{c_1, c_2, \ldots, c_m\}$. In our work, passages P_i are collected from Wikipedia and retrieved by an external retriever according to Q_i. One passage p_i may include one or more sentences. Each citation in C_i is an index of one retrieved passage p_i. We show an example pair (Q_i, P_i, R_i, C_i) in Table 1.

2.2 Overview of Our Framework

To construct an effective attribution system at a low cost, we introduce the framework $\mathcal{A}^3$, which consists of a data construction process and a model training process. An illustration of our framework is shown in Figure 1. In the data construction process, we focus on generating a training set $D_{train} = \{d_i\}$ for attributed text generation, where d_i represents (Q_i, P_i, R_i, C_i). Due to the high cost of human annotation, our framework aims to automatically generate $D = \{d_i\}$ based on existing corpus $D_{corpus} = \{p_1, p_2, \ldots, p_N\}$ (such as Wikipedia), where N denotes the number of passages in D_{corpus}. Specifically, during data construction, we generate (Q_i, R_i, C_i) based on the P_i. We introduce quite a few strategies to ensure the generated data quality for the Supervised Fine-Tuning (SFT) and Preference Optimization (PO) stages. Next, we introduce the data construction process of both the SFT and PO stages and the model training process. The example of our generated dataset is shown in Table 1.

2.3 Data Construction for Supervised Fine-tuning

In the SFT stage, to generate an SFT training sample d_i, we first select interrelated documents $P_i = \{p_1, p_2, \ldots, p_n\}$ from D_{corpus} with the use of an entity matching based document clustering strategy, where n denotes the number of selected documents. Then, we use LLMs to generate (Q_i, R_i, C_i) based on the selected P_i. Finally, a data filtering strategy and a noisy document aug-

Query:	When did the Battle of Rennell Island occur and why is it significant in the context of World War II's Guadalcanal campaign?
Retrieved Passages:	Document [1] The Battle of Rennell Island took place on 29 – 30 January 1943 . . . . Document [2] . . . it was the last major naval engagement between the United States Navy and the Imperial Japanese Navy during the Guadalcanal campaign of World War II . . . Document [3] . . . Document [4] = = Goalball = = Rzepecki is a goalball player , . . . Document [5] Aviva Premiership rugby union teams are based in London , . . .
Preferred Response:	The Battle of Rennell Island occurred on 29-30 January 1943 and is significant because it was the last major naval engagement between the United States Navy and the Imperial Japanese Navy during the Guadalcanal campaign of World War II [1][2][3].
Dispreferred Response:	The Battle of Rennell Island occurred on 29-30 January 1943 and is significant because it was the last major naval engagement between the United States Navy and the Imperial Japanese Navy during the Guadalcanal campaign of World War II [1]~~[2]~~[3][5].

Table 1: A example of our generated dataset. In real-world scenarios, retrieval systems inevitably retrieve irrelevant documents. Therefore, to simulate real-world scenarios and reduce the effect of irrelevant documents, we select random passages as noisy retrieved passages, which are shown as Gray. Red represents wrong modifications caused by our preference data construction strategies.

mentation strategy are introduced to enhance the SFT data quality.

2.3.1 Interrelated Document Selection

There are two alternatives for the interrelated document selection. The first and simpler one is that if the number of selected documents $n = 1$, we pick up one passage p_i from D_{corpus} randomly. For the other one, we need to select more than two documents from D_{corpus} for (Q_i, P_i, R_i, C_i) pair construction. This phenomenon results from that the randomly selected documents are not interrelated. Therefore, we introduce an entity matching based document clustering strategy to select interrelated documents. Specifically, we use WikiGraphs (Wang et al., 2021) as D_{corpus}. WikiGraphs consists of lots of entity edges $Edge_i = (entity_a, entity_b, r)$, which represents $entity_a$ has relation r to $entity_b$. Besides, $entity_a$ and $entity_b$ links to P_a and P_b respectively, where P_a and P_b represents Wikipedia passages $\{p_1, p_2, \ldots, p_n\}$. Focusing on a specific entity, we gather a set of Wikipedia passages linked to entities that have a relationship r with our target entity. To further uncover interconnected paragraphs within these passages, we seek out sections that feature more than two entities as interrelated documents $P_i = \{p_1, p_2, \ldots, p_n\}$.

2.3.2 Q&A Pair Extraction

After selecting interrelated documents $P_i = \{p_1, p_2, \ldots, p_n\}$, we use *gpt-4-1106-preview* to generate attributed Q&A pair (Q_i, R_i, C_i). If the number of selected documents $n = 1$, recognizing that generating a question for a given response is more straightforward than answering a question, we use LLMs to generate a summary for this document p_1 and treat this summary as response R_i. Subsequently, we task the LLMs with generating question Q_i based on R_i. Afterward, we add the citation c_1 linked with p_1 as C_i. If the number of selected documents $n \geq 2$, to improve the speed of data construction, we generate multiple attributed Q&A pairs based on P_i. The model will output multiple pairs of Q&A, and we extract them using regular expressions. We show the prompt templates in the appendix.

2.3.3 Data Filtering

To conduct data filtering, we remove generated samples with low citation quality. We choose the citation quality criterion introduced from ALCE (Gao et al., 2023b) to evaluate the citation quality for each generated sample (Q_i, P_i, R_i, C_i). Specifically, each response is divided into multiple statements and the NLI model[4] is used to determine whether each statement is fully supportive or not fully supportive. Citation recall is a metric to evaluate whether the cited passages fully support the content of the response, which is calculated by the average support ratio of all the claims in the response. Citation precision is employed to identify irrelevant citations. A citation becomes irrelevant to a statement when it fails to substantiate the statement, yet the remaining citations continue to support the statement without it. Citation precision is calculated by the average relevant ratio of all the citations in the response. Finally, we remove samples whose Citation F1 is below a threshold, which

[4]We use `https://huggingface.co/google/t5_xxl_true_nli_mixture`.

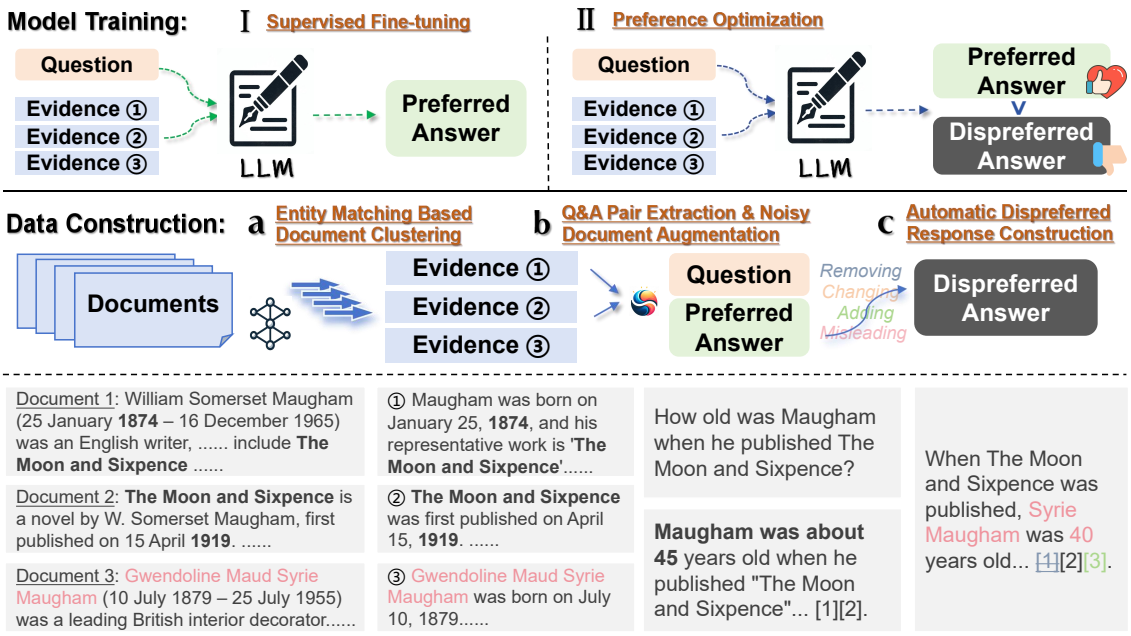

Figure 1: An illustration of our $\mathcal{A}^3$ framework, including two main processes: **Data Construction** and **Model Training**. During the data construction process, we first introduce an **(a) entity matching based document clustering** strategy to select multiple interrelated documents. Next, we proceed with **(b) Q&A Pair Extraction & Noisy Document Augmentation** using the selected documents to generate Q&A pairs that serve as the SFT training data. Following this step, **(c) Automatic Dispreferred Response Construction** involves using the responses from **(b)** as preferred responses and altering these responses by removing, changing, or adding citations, or by emphasizing irrelevant documents to construct dispreferred responses. During the model training process, we utilize the constructed data to perform **(I) Supervised Fine-tuning** and **(II) Preference Optimization**.

is computed as follows:

$$\text{Citation F1} = 2 \times \frac{\text{Citation Precision} \times \text{Citation Recall}}{\text{Citation Precision} + \text{Citation Recall}}$$

2.3.4 Noisy Document Augmentation

Due to the limitation of retrieval systems, the retrieved passages are difficult to avoid containing some irrelevant information. To address the above challenge, we introduce a noisy document augmentation strategy. Specifically, we first select some random documents from D_{corpus} as irrelevant documents and add them to the documents set $\{p_1, p_2, \ldots, p_n\}$. Then we shuffle the order of the final documents set P_i and change $\{c_1, c_2, \ldots, c_m\}$ to ensure they link with the correct passage.

2.4 Data Construction for Preference Optimization

In the PO stage, we use R_i as the preferred response and design four strategies to automatically generate corresponding disprefered response DR_i with incorrect citations DC_i:

- Random Citation Adding: To prevent LLMs from including excessive incorrect citations, we add some citations randomly to construct a dispreferred response;
- Random Citation Removing: We remove some golden citations to construct a dispreferred response, which aims to avoid the miss of key citations;
- Random Citation Changing: To avoid referencing irrelevant documents, we substitute some key citations with random ones as a dispreferred response;
- Irrelevant Document Focusing: To discourage LLMs from focusing on irrelevant documents, we first remove documents related to responses and retain only irrelevant documents. Then, we use LLMs to answer the given questions based on these irrelevant documents, employing these answers as dispreferred responses.

2.5 Model Training

2.5.1 Supervised Fine-tuning Stage

We recognize each sample as follows:

Instruction: Write a high-quality answer for the given question using only the provided search results and cite them properly using [1][2][3].

Question: *{Q_i}*

Document[j]:{p_j}

Output:{R_iC_i}

We only compute cross-entropy loss on output to train the SFT modelM_{SFT}.

2.5.2 Preference Optimization Stage

In the PO stage, we use Direct Preference Optimization (DPO) (Rafailov et al., 2023) loss function. We use the prompt template the same as the one used in the SFT stage. Formally, we use $\mathcal{L}_{DPO}$ to train the DPO model M_{DPO}:

$$\mathcal{L}_{DPO} = -\mathbb{E}\left[\log\sigma\left(\alpha\frac{M_{DPO}(R_iC_i|Q_iP_i)}{M_{SFT}(R_iC_i|Q_iP_i)} - \alpha\frac{M_{DPO}(DR_iDC_i|Q_iP_i)}{M_{SFT}(DR_iDC_i|I)}\right)\right]$$

where σ represents the logistic function, α is a hyper-parameter.

3 Experiments

3.1 Backbone and Baselines

To confirm the generalization of our framework, we select two families of LLMs as backbone models: LLaMA2 (Touvron et al., 2023) and Mistral (Jiang et al., 2023). For the LLaMA2 family, we select two models with different sizes: LLaMA2-7B and LLaMA2-13B. For the Mistral family, we select Mistral-7B. We use ChatGPT (gpt-3.5-turbo-0301) with a 4K context window for most main experiments and ablations. We also report results with ChatGPT-16K (gpt3.5-turbo-16k-0613) and GPT-4 (gpt-4-0613;8K context window). For open-source models, we evaluate LLaMA and its chat versions. Besides, we also compare our framework with Hagrid (Kamalloo et al., 2023), Self-RAG (Asai et al., 2024a), and CaLM (Hsu et al., 2024). Hagrid (Kamalloo et al., 2023) is an SFT dataset for attributed text generation, which consists of 3,214 samples; Self-RAG (Asai et al., 2024a) aims to make LLMs decide whether need to retrieve documents through self-reflection; CaLM (Hsu et al., 2024) empowers smaller LMs to validate the output of larger LMs.

3.2 Benchmark

To fully confirm the effectiveness, we evaluate our proposed framework and all the baselines on one short-form QA dataset PopQA (Mallen et al., 2023) and two long-form QA datasets ALCE-ASQA and ALCE-ELI5 (Gao et al., 2023b). Since our data is generated from Wikipedia passages, while ELI5 is collected from Reddit, the performance of the ELI5 can dispel the concern about the performance improvements gained from potential data contamination. For ASQA, we use EM recall, citation recall, and precision; for ELI5, we use claim, citation recall, and precision. Following Self-RAG (Asai et al., 2024a), we utilize 1,399 long-tail entity queries for PopQA and evaluate performance based on whether gold answers are included in the model generations rather than exact matching.

3.3 Dataset Statistics

Our generated dataset consists of 13,225 samples, each containing a question, retrieved passages, preferred responses, and dispreferred responses. The average length of questions is 17.89 words, preferred responses are 32.00 words, and dispreferred responses are 30.59 words. The example of our dataset is shown in Table 1.

3.4 Main Results

In Table 2, we report the results of our framework and all the baselines. With the help of our framework, the open-source backbone models can achieve strong citation performance, which can outperform powerful closed-sourced models significantly. Besides, our framework can bring significant improvements to all the backbone models on all three datasets, which fully shows the generalization of $\mathcal{A}^3$. The ablation study of the SFT and PO stages also shows the effectiveness of our strategies for each stage. PO stages can bring significant improvements in the caution quality. Depite the strong citation performance, the correctness of our framework is below powerful closed-sourced models. We think this phenomenon may result from the limitation of the model size.

3.5 The Effect of Document Selection Strategy

In our preliminary experiments, we find that selecting multiple documents randomly as P_i to generate (Q_i, R_i, C_i) causes the generated response to usually have only one citation. We speculate this phenomenon results from the selected documents that are not interrelated. Therefore, in this experiment, we study the effect of different document selection strategies. We compare the entity matching based document clustering strategy with the query

Method	ASQA			ELI5			PopQA
	Correct	Citation		Correct	Citation		Correct
	EM Rec.	Rec.	Prec.	Claim	Rec.	Prec.	Acc.
LLaMA2-7B (Touvron et al., 2023)	12.4	1.3	2.7	2.4	0.8	0.9	45.9
LLaMA2-7B-Hagrid (Kamalloo et al., 2023)	28.5	48.8	54.2	9.7	18.8	30.2	40.7
LLaMA2-7B-$\mathcal{A}^3$-SFT (*Ours*)	31.3	73.6	71.4	9.4	33.6	33.9	49.5
LLaMA2-7B-$\mathcal{A}^3$-PO (*Ours*)	33.2	80.7	76.1	8.6	44.7	46.3	52.1
LLaMA2-13B (Touvron et al., 2023)	26.9	10.6	15.4	3.9	3.1	5.3	21.9
LLaMA2-13B-Vicuna (Chiang et al., 2023)	31.9	51.1	50.1	10.0	15.6	19.6	-
LLaMA2-13B-Chat (Touvron et al., 2023)	35.2	38.4	39.4	13.4	17.3	15.8	-
LLaMA2-13B-Hagrid (Kamalloo et al., 2023)	28.7	46.5	47.0	7.9	14.8	17.4	27.6
LLaMA2-13B-Self-RAG (Asai et al., 2024a)	31.7	70.3	71.3	10.7	20.8	22.5	-
LLaMA2-13B-$\mathcal{A}^3$-SFT (*Ours*)	31.5	79.9	79.6	9.1	41.6	41.3	45.7
LLaMA2-13B-$\mathcal{A}^3$-PO (*Ours*)	31.7	82.3	82.5	8.9	42.0	42.3	47.1
LLaMA3.1-8B (Dubey et al., 2024)	27.2	2.4	4.1	7.7	2.5	5.3	17.3
LLaMA3.1-8B-Hagrid (Kamalloo et al., 2023)	28.6	49.7	50.3	6.9	17.6	19.3	30.9
LLaMA3.1-8B-$\mathcal{A}^3$-SFT (*Ours*)	31.3	80.0	79.8	7.6	40.4	40.2	35.8
LLaMA3.1-8B-$\mathcal{A}^3$-PO (*Ours*)	32.5	82.7	83.2	7.4	48.8	51.8	50.0
Mistral-7B (Jiang et al., 2023)	13.3	0.7	1.7	3.8	1.4	2.7	42.3
Mistral-7B-Hagrid (Kamalloo et al., 2023)	15.2	36.1	40.1	5.4	20.2	28.2	43.7
Mistral-7B-$\mathcal{A}^3$-SFT (*Ours*)	31.4	76.8	75.6	8.6	37.6	36.9	47.2
Mistral-7B-$\mathcal{A}^3$-PO (*Ours*)	31.7	**84.4**	**87.0**	6.0	**60.7**	**68.9**	**52.8**
LLaMA2-70B-Chat (Touvron et al., 2023)	41.5	62.9	61.3	12.8	38.3	37.9	-
GPT-3.5-Turbo (OpenAI, 2023b)	40.4	73.6	72.5	12.0	51.1	50.0	-
GPT-4 (OpenAI, 2023a)	41.3	73.0	76.5	**14.2**	48.5	53.4	-
GPT-3.5-Turbo-CaLM (Hsu et al., 2024)	**45.0**	78.0	72.6	12.9	51.9	46.6	-

Table 2: The performance of all the baseline and our proposed framework on ASQA, ELI5, and PopQA. ***Bold*** indicates the best performance. **EM**, **Rec.**, **Prec.** and **Acc.** denote Exact Match, Recall, Precision, and Accuracy.

Document Selection	Whether QG	DG Model	Correct	Citation	
			EM Rec.	Rec.	Prec.
Query Based	✗	`gpt-3.5-turbo-0301`	28.7	46.5	47.0
Query Based	✗	`gpt-4-1106-preview`	29.9	62.6	55.2
Query Based	✓	`gpt-4-1106-preview`	30.8	73.0	51.9
Entity Based	✓	`gpt-4-1106-preview`	**31.3**	**73.2**	**72.7**

Table 3: The effect of different document selection and data generation strategies. ***Bold*** indicates the best performance. **Whether QG** represents whether we generate questions based on the selected documents. **DG Model** denotes the LLM we used to generate data.

based document clustering strategy. Specifically, we utilize a retriever to get multiple passages related to one query and treat these retrieved passages as selected passages. We collect these queries and retrieved passages from MIRACL (Zhang et al., 2022). For the query based document clustering strategy, since we have achieved queries, we also compare the performance whether using this query as Q_i or generating Q_i as the entity matching based document clustering strategy. The results in Table 3 confirm the effectiveness of our proposed entity matching based document clustering strategy. Besides, we can find that the data generation model has a significant effect on the performance.

3.6 The Effect of Data Filtering and Noisy Document Augmentation Strategies

To analyze our framework further, we also study the effect of data filtering and noisy document augmentation strategies. From the results in Figure 2, we can find that only a high filtering threshold can bring benefits, which shows that data quality is more important than data number. Besides, we also find that significant contribution of these two strategies to avoiding unnecessary, irrelevant citations. As shown in Figure 3, the noisy document augmen-

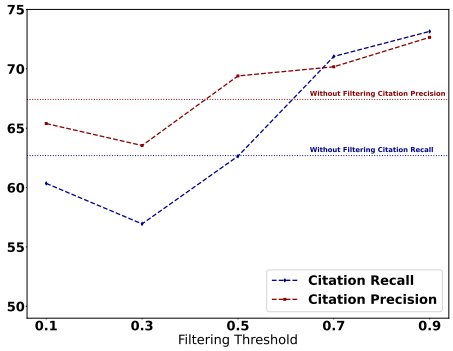

Figure 2: The effect of the filtering threshold in the SFT stage. We use Citation F1 score as filtering threshold and report the performance on ASQA when using LLaMA2-7B as backbone models.

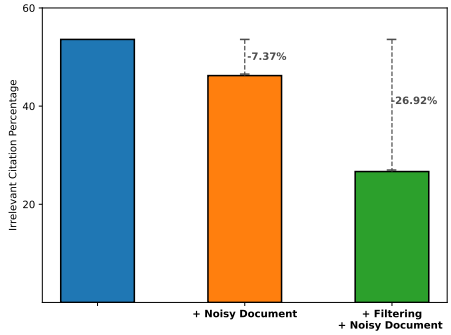

Figure 3: The irrelevant citation percentage when using Data Filtering and Noisy Document Augmentation strategies. We report the performance on ASQA when using LLaMA2-7B as backbone models.

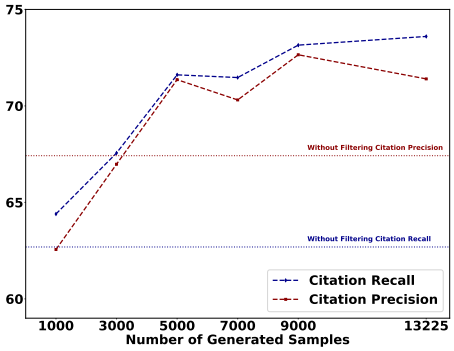

Figure 4: The effect of the data number in the SFT stage. We report the performance on ASQA when using LLaMA2-7B as backbone models.

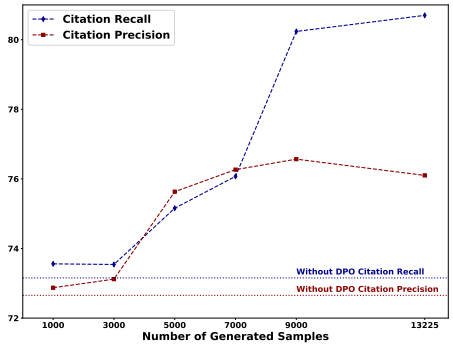

Figure 5: The effect of the data number in the PO stage. We report the performance on ASQA when using LLaMA2-7B as backbone models.

tation and data filtering can reduce the percentage of irrelevant citations.

3.7 Data Filtering&Number Analysis

We report the results when generating different numbers of data in Figure 4 and 5. With the same filtering threshold, the performance is improved with the increasing of the SFT data, which is shown in Figure 4. When the SFT data number is more than 5,000, the improvements are not significant. In Figure 5, we can find that when the DPO data number is less than 9,000, there is a clear linear relationship, namely, as the DPO data increases, performance improves, which fully confirms the effectiveness of our proposed framework. When the DPO data number is more than 9,000, the improvements are not significant.

3.8 Case Study

In Table 4, we show results predicted by LLaMA2-7B-A3-PO on the ASQA dataset. We can observe that the response is fluent, relevant to the question, and contains proper citations, confirming the effectiveness of our framework.

3.9 Human Evaluation

We conduct a human evaluation on 100 randomly selected samples from our dataset. Following Self-RAG (Asai et al., 2024a), we evaluate from two dimensions: relevance (the output's appropriateness and topic alignment with the question) and supportiveness (the sufficiency of provided evidence for answer validation). The results show that 84% of the samples were classified as relevant and 78% as supportive. These results confirm the high quality of our dataset.

4 Related Work

LLM Alignment Existing LLM alignment works mainly consist of two methods: supervised instruction tuning and reinforcement learning. Supervised instruction tuning leverages collected supervised instruction datasets to train foundation models to deal with diverse instructions.

Input:	Write a high-quality answer for the given question using only the provided search results and cite them properly using [1][2][3]. Question: Who wrote the song i'm not going to miss you? Document [1](Title: I'm Not Gonna Miss You): part of the soundtrack for "" which premiered on October 24, 2014. I'm Not Gonna Miss You "I'm Not Gonna Miss You" is a song recorded by . . . Document [2](Title: I'm Not Gonna Miss You): I'm Not Gonna Miss You "I'm Not Gonna Miss You" is a song recorded by American country music artist Glen Campbell and The Wrecking Crew. . . Document [3](Title: Don't Miss You (Ricki-Lee Coulter song)): Don't Miss You (Ricki-Lee Coulter song) "Don't Miss You" is a song performed by Australian singer-songwriter, Ricki-Lee Coulter. . . Document [4](Title: I'm Not Gonna Miss You): Grammy Award for Best Country Song. "I'm Not Gonna Miss You" was initiated by Julian Raymond who pitched an idea of recording the song to James Keach. . . Document [5](Title: Don't Miss You (Ricki-Lee Coulter song)): physical release of "Don't Miss You", Coulter held instore appearances at Westfield Shopping Centres in Victoria, New South Wales. . . Answer:
Output	The song "I'm Not Gonna Miss You" was co-written by Glen Campbell and producer Julian Raymond [1][2].

Table 4: A example of responses generated by LLaMA2-7B-A3-PO on the ASQA dataset.

Early instruction datasets are collected from large-scale existing NLP task datasets and transformed into instruction formats with manual written templates (Wei et al., 2021). In order to align with human requirements in realistic scenarios, recent works focus on collecting instruction data from realistic scenarios (Ouyang et al., 2022; Databricks, 2023; Köpf et al., 2023; Zhang et al., 2023a; Chiang et al., 2023). The reinforcement learning method improves the response quality with the preference signals. InstructGPT (Ouyang et al., 2022) introduce a Proximal Policy Optimization (PPO) framework, which can help LLMs learn from human preference signals. There is a line of subsequent work focused on improving the effectiveness and efficiency of this framework, e.g., RAFT (Dong et al., 2023), DPO (Rafailov et al., 2023), PRO (Song et al., 2023), COH (Liu et al., 2023) and RRHF (Yuan et al., 2023).

Attributed LLM Recently, retrieval-augmented models have shown great potential. By utilizing retrieved passages to prompt models, they can enhance the correctness of the outputs, which have been applied to various downstream tasks. To further improve correctness and help users more easily verify the outputs, recent works focus on building attributed LLM, which generates text with citations. Due to the abundant knowledge obtained during the pre-training stage, there is a line of works that focus on exploring the potential of LLMs to generate citation directly (Weller et al., 2023; Xu et al., 2023; Asai et al., 2024a). To test the performance of attributed text generation, recent works build specific datasets for attribution. Specifically, CiteBench (Funkquist et al., 2022) focus on text summarization; ALCE (Gao et al., 2023b) is a comprehensive benchmark to evaluate the attribution ability of existing LLMs, which do not contain train data; HAGRID (Kamalloo et al., 2023) utilize powerful LLMs (GPT-3.5-turbo) to generate texts with citations. BioKaLMA (Li et al., 2023b) and ExpertQA (Malaviya et al., 2023) are specific-domain datasets for attribution. Besides, rather than generating citations directly, there is also a line of works (Gao et al., 2023a; Huo et al., 2023; Chen et al., 2023; Hsu et al., 2024) perform retrieving and finding supportive passages to add citations after generating outputs. Recent works utilize reward modeling (Huang et al., 2024), test-time adaptation (Ye et al., 2024), or preference learning (Li et al., 2024) to improve the performance of attribution LLM. However, all these works construct their training sets by carefully collecting existing supervised datasets. In contrast, our framework aims to generate high-quality attributed query-response pairs without requiring extensive human-annotated samples. Additionally, we believe our method could be combined with these existing works to bring improvements further.

5 Conclusion

In this paper, we introduce the $\mathcal{A}^3$ framework, which aims to leverage underperforming attributed LLMs to generate high-quality attributed query-response pairs for both SFT and PO stages, eliminating the requirement for human-annotated samples. Comprehensive experiments demonstrate our method's effectiveness in improving citation quality and reducing the effect of irrelevant documents. Despite the effectiveness of $\mathcal{A}^3$, there remains a gap in fully meeting the needs of information-seeking tasks in realistic scenarios. With our framework, we hope more breakthroughs can be made to promote the development of this task.

Limitations

Although our instruction framework can generate high-quality attributed text generation data, it still has the following limitations:

- Even though we generate data with the use of the existing corpus to ensure the faithfulness of outputs, due to the characteristic of the used backbone LLMs and the potential social bias in the corpus, it may still have the hallucination and bias problem. Therefore, the generated and data and the outputs of trained models may contain misleading and toxic information, which needs to be addressed before being applied to realistic scenario;

- Although we conduct experiments on widely used attributed text generation benchmarks, the language of all these benchmarks is English, which has limited morphology. The effectiveness of our proposed method on language with varied morphology needs to be further confirmed.

- We have confirmed the effectiveness of our framework on multiple backbone LLMs, including LLaMA2-7B, LLaMA2-13B, LLaMA3.1-8B and LLaMA3.1-8B. However, the effectiveness on other backbone LLMs still need to be tested.

- In our framework, we mainly use GPT-3.5-Turbo and GPT-4 to generate data, which can be expanded to include other closed-source or open-source LLMs, such as Claude and LLaMA3.1-Instruct.

References

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2024a. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations*.

Akari Asai, Zexuan Zhong, Danqi Chen, Pang Wei Koh, Luke Zettlemoyer, Hannaneh Hajishirzi, and Wen-tau Yih. 2024b. Reliable, adaptable, and attributable language models with retrieval. *arXiv preprint arXiv:2403.03187*.

Jifan Chen, Grace Kim, Aniruddh Sriram, Greg Durrett, and Eunsol Choi. 2023. Complex claim verification with evidence retrieved in the wild. *arXiv preprint arXiv:2305.11859*.

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

Databricks. 2023. Databricks' dolly, a large language model trained on the databricks machine learning platform. https://github.com/databrickslabs/dolly.

Hanze Dong, Wei Xiong, Deepanshu Goyal, Rui Pan, Shizhe Diao, Jipeng Zhang, Kashun Shum, and Tong Zhang. 2023. Raft: Reward ranked finetuning for generative foundation model alignment. *arXiv preprint arXiv:2304.06767*.

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*.

Martin Funkquist, Ilia Kuznetsov, Yufang Hou, and Iryna Gurevych. 2022. Citebench: A benchmark for scientific citation text generation. *arXiv preprint arXiv:2212.09577*.

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, Singapore. Association for Computational Linguistics.

I Hsu, Zifeng Wang, Long T Le, Lesly Miculicich, Nanyun Peng, Chen-Yu Lee, Tomas Pfister, et al. 2024. Calm: Contrasting large and small language models to verify grounded generation. *arXiv preprint arXiv:2406.05365*.

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

Siqing Huo, Negar Arabzadeh, and Charles Clarke. 2023. Retrieving supporting evidence for generative question answering. In *Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region*, pages 11–20.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and

Edouard Grave. 2022. Unsupervised dense information retrieval with contrastive learning. *Transactions on Machine Learning Research*.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Ehsan Kamalloo, Aref Jafari, Xinyu Zhang, Nandan Thakur, and Jimmy Lin. 2023. Hagrid: A human-llm collaborative dataset for generative information-seeking with attribution. *arXiv preprint arXiv:2307.16883*.

Andreas Köpf, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi-Rui Tam, Keith Stevens, Abdullah Barhoum, Nguyen Minh Duc, Oliver Stanley, Richárd Nagyfi, et al. 2023. Openassistant conversations–democratizing large language model alignment. *arXiv preprint arXiv:2304.07327*.

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

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

Xinze Li, Yixin Cao, Liangming Pan, Yubo Ma, and Aixin Sun. 2023b. Towards verifiable generation: A benchmark for knowledge-aware language model attribution. *arXiv preprint arXiv:2310.05634*.

Hao Liu, Carmelo Sferrazza, and Pieter Abbeel. 2023. Languages are rewards: Hindsight finetuning using human feedback. *arXiv preprint arXiv:2302.02676*.

Chaitanya Malaviya, Subin Lee, Sihao Chen, Elizabeth Sieber, Mark Yatskar, and Dan Roth. 2023. Expertqa: Expert-curated questions and attributed answers. *arXiv preprint arXiv:2309.07852*.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9802–9822.

Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernandez Abrego, Ji Ma, Vincent Zhao, Yi Luan, Keith Hall, Ming-Wei Chang, and Yinfei Yang. 2022. Large dual encoders are generalizable retrievers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9844–9855, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

OpenAI. 2023a. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

OpenAI. 2023b. Introducing chatgpt.

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.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D Manning, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *arXiv preprint arXiv:2305.18290*.

Feifan Song, Bowen Yu, Minghao Li, Haiyang Yu, Fei Huang, Yongbin Li, and Houfeng Wang. 2023. Preference ranking optimization for human alignment. *arXiv preprint arXiv:2306.17492*.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Luyu Wang, Yujia Li, Ozlem Aslan, and Oriol Vinyals. 2021. Wikigraphs: A wikipedia text-knowledge graph paired dataset. *NAACL-HLT 2021*, page 67.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

Orion Weller, Marc Marone, Nathaniel Weir, Dawn Lawrie, Daniel Khashabi, and Benjamin Van Durme. 2023. " according to..." prompting language models improves quoting from pre-training data. *arXiv preprint arXiv:2305.13252*.

Shicheng Xu, Liang Pang, Huawei Shen, Xueqi Cheng, and Tat-seng Chua. 2023. Search-in-the-chain: Towards the accurate, credible and traceable content generation for complex knowledge-intensive tasks. *arXiv preprint arXiv:2304.14732*.

Xi Ye, Ruoxi Sun, Sercan Arik, and Tomas Pfister. 2024. 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.

Zheng Yuan, Hongyi Yuan, Chuanqi Tan, Wei Wang, Songfang Huang, and Fei Huang. 2023. Rrhf: Rank responses to align language models with human feedback without tears. *arXiv preprint arXiv:2304.05302*.

Ge Zhang, Yemin Shi, Ruibo Liu, Ruibin Yuan, Yizhi Li, Siwei Dong, Yu Shu, Zhaoqun Li, Zekun Wang, Chenghua Lin, et al. 2023a. Chinese open instruction generalist: A preliminary release. *arXiv preprint arXiv:2304.07987*.

Xinyu Zhang, Nandan Thakur, Odunayo Ogundepo, Ehsan Kamalloo, David Alfonso-Hermelo, Xiaoguang Li, Qun Liu, Mehdi Rezagholizadeh, and Jimmy Lin. 2022. Making a miracl: Multilingual information retrieval across a continuum of languages. *Preprint*, arXiv:2210.09984.

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, et al. 2023b. Siren's song in the ai ocean: A survey on hallucination in large language models. *arXiv preprint arXiv:2309.01219*.

A Prompt Template

In Q&A Pair Extraction,we generate R_i based on P_i using the following prompt template:

> ***I will give a reference paragraph. Please summarize this paragraph briefly.***
>
> ***Reference:{P_i}***
>
> ***Summary:***

We generate Q_i based on R_i using the following prompt template:

> ***I will give an answer. Please design a question for this answer.***
>
> ***Answer:{R_i}***
>
> ***Question:***

We generate multiple attributed Q&A pairs based on P_i with the use of following prompt template:

> ***I will give some reference paragraphs. Please design some question-answer pairs based on these paragraphs. Each question starts with Q: and each answer starts with A:. You should consider the interconnectedness of content across multiple paragraphs and formulate questions that draw connections between the information presented in those paragraphs. Also, mention the reference of parts of your answer based on the given paragraphs within brackets [] as in the IEEE format.***
>
> ***Reference:{P_i}***

B Implementation Details

B.1 Training Details

We implement all models with the open-source toolkit *Transformers* [5]. In the SFT stage, we follow Alpaca (Taori et al., 2023) use AdamW optimizer and set the learning rate to 1e-5, batch size to 32, learning rate warmup ratio to 0.03, input length to 2,048 and perform training for 3 epochs. We update all parameters during the fine-tuning stage. Our DPO implementation is based on the Alignment-Handbook [6]. In the DPO stage, we set the learning rate to 5e-7, batch size to 16, learning rate warmup ratio to 0.1, input length to 2,048, and perform training for 1 epoch. All the fine-tuning experiments are performed with 80GB NVIDIA A100 GPUs.

B.2 Inference Details

During the model inference state, we use the prompt template the same as the one used in the SFT stage. We use the same retriever and documents with Top-5 scores for our method and all the baselines. Specifically, following ALCE (Gao et al., 2023b) and Self-RAG (Asai et al., 2024a), we use GTR (Ni et al., 2022) for ASQA, BM25 for ELI5 and Contriever-MS MARCO (Izacard et al., 2022) for PopQA.

[5] https://github.com/huggingface/transformers

[6] https://github.com/huggingface/alignment-handbook