Improving Attributed Text Generation of Large Language Models via Preference Learning

Dongfang Li, Zetian Sun, Baotian Hu*,
Zhenyu Liu, Xinshuo Hu, Xuebo Liu, Min Zhang
Harbin Institute of Technology (Shenzhen), Shenzhen, China
{lidongfang,hubaotian,zhangmin2021}@hit.edu.cn

Abstract

Large language models have been widely adopted in natural language processing, yet they face the challenge of generating unreliable content. Recent works aim to reduce misinformation and hallucinations by resorting to attribution as a means to provide evidence (i.e., citations). However, current attribution methods usually focus on the retrieval stage and automatic evaluation that neglect mirroring the citation mechanisms in human scholarly writing to bolster credibility. In this paper, we address these challenges by modeling the attribution task as preference learning and introducing an Automatic Preference Optimization (APO) framework. First, we create a curated collection for post-training with 6,330 examples by collecting and filtering from existing datasets. Second, considering the high cost of labeling preference data, we further propose an automatic method to synthesize attribution preference data resulting in 95,263 pairs. Moreover, inspired by the human citation process, we further propose a progressive preference optimization method by leveraging fine-grained information. Extensive experiments on three datasets (i.e., ASQA, StrategyQA, and ELI5) demonstrate that APO achieves state-of-the-art citation F1 with higher answer quality. [1]

1 Introduction

Large Language Models (LLMs) have demonstrated emergent abilities and have gained widespread application in Natural Language Processing (NLP) (Brown et al., 2020; Wei et al., 2022; OpenAI, 2022; Anil et al., 2023). For example, LLMs have shown remarkable in-context learning capabilities across a variety of domains and tasks (Dong et al., 2023). Although LLMs have been widely adopted, a prominent issue is that they produce hallucinations in certain situations (Ye et al., 2023a; Zhang et al., 2023). In other words, they generate information that sounds plausible but is nonfactual, thereby limiting their applicability in the real world. To mitigate hallucinations, researchers have resorted to grounding statements in responses generated by LLMs to supported evidence, either by providing rationales or by adding citations to the statements (Li et al., 2023a; Liu et al., 2023).

Recent works have utilized external knowledge sources such as retrieved documents and knowledge graphs for attribution (Shuster et al., 2021; Li et al., 2023c). Generally, these works are divided into two types: 1) the model generates an answer with citations based on the retrieved documents (Li et al., 2023b); 2) an answer is first generated, then modified again to add attribution references by retrieving with query and initial answer (Gao et al., 2023a). However, these works focus mainly on the retrieval stage (Ye et al., 2023b) and the evaluation process (Yue et al., 2023). Considering the selection of the model's desired responses and behavior from its very broad knowledge and capabilities, it is more necessary to optimize the generation process, not only reducing the hallucination of the original answer but also avoiding the hallucination of the attribution process. On the other hand, fine-tuning LLMs after pre-training can also significantly improve performance for users' downstream tasks. First, given positive examples of correct behavior, supervised fine-tuning can be performed using standard likelihood-based training. Secondly, given positive and negative examples (binary feedback or pairwise feedback), methods such as unlikelihood training on negative examples (Welleck et al., 2020) or RLHF-PPO (Ziegler et al., 2019) can be used for learning. However, these methods usually suffer from expensive data collection process, reward model training, sparse reward and text degeneration problems, making them difficult to use

*Corresponding author.
[1]Code is released in https://github.com/HITsz-TMG/ATG-PO

Findings of the Association for Computational Linguistics ACL 2024, pages 5079–5101

in practical applications (Azar et al., 2023).

In this paper, inspired by the citation mechanisms in human scholarly writing (Brooks, 1986; Teplitskiy et al., 2022), we address these challenges by conceptualizing the attribution task for LLMs as preference learning and proposing an Automatic Preference Optimization (APO) framework, as shown in Figure 1. Initially, we assemble a curated dataset comprising 6,330 examples sourced and refined from existing datasets for post-training. This step makes the LLMs know the basic format and requirements of attribution. Considering the substantial cost and extremely time-consuming of preference pair annotations, we thus introduce an automated approach to generate attribution preference data, yielding 95,263 pairs. Furthermore, drawing inspiration from the human process of citation and direct preference optimization (Rafailov et al., 2023), we propose a progressive preference optimization method with experience replay bypassing the need for explicit reward modeling or reinforcement learning. We conduct the extensive experiment on three datasets (i.e., ASQA, StrategyQA, and ELI5). The experiment results demonstrate that APO surpasses compared baselines across all datasets with improved citation F1 along with higher response quality. Our contributions are summarized as follows:

- To the best of our knowledge, we are the first to apply preference learning for attribution tasks. We also show that our method can be applied under synthesized preference scenarios.
- We establish a full data collection pipeline for attribution tasks and will open-source our all authorized data after publication for future research.
- We propose a progressive preference optimization method to alleviate the sparse reward problem by leveraging fine-grained information. We further benchmark existing direct preference optimization methods and provide insights for attribution tasks.

2 Related Work

2.1 Text Generation for Verification

Prior works have studied methods and evaluations for verification that identify supporting sources for model outputs. For instance, Rashkin et al. (2021) introduce the concept of Attributable to Identified Sources (AIS) which transforms model outputs into standalone, interpretable propositions. The response s can be attributed to a source P if they meet the intuitive criterion "*According to P, s*". Bohnet et al. (2022) adapt the AIS framework for QA scenarios. Further, Gao et al. (2023b) extrapolate AIS to evaluate generated text of LLMs with citations. Additionally, several works focus on building and using automated AIS evaluations (Honovich et al., 2022; Gao et al., 2023a; Liu et al., 2023). For a comprehensive overview, please refer to Li et al. (2023a). In contrast to existing approaches, our work broadens the scope of attribution beyond just verifiable text generation and devises a methodology to enhance these attributions which frames it as a preference learning problem.

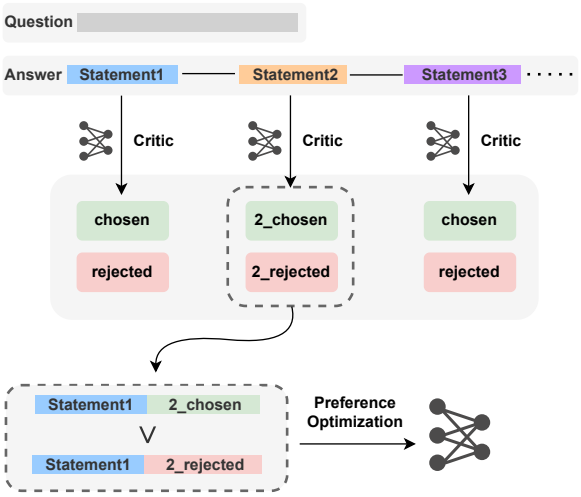

Figure 1: A brief overview of our APO framework. For some given question, model first generates an answer with several statements. Then for each statement, we regenerate two parallel statements (the positive one and the negative one, given a specific error type). Then we preform preference optimization in the statement level. The "2_chosen" means it's the second statement in the answer that be selected to construct preference pairs. We show APO in more detail in Figure 2.

2.2 Preference Optimization Methods

Preference Optimization (PO) methods significantly improve generate quality to align with human values (Christiano et al., 2017; Ziegler et al., 2019; Stiennon et al., 2020; Bai et al., 2022). It usually first collects pairs of generations under the same context and a pairwise human preference to indicate which generation is better. Then the PO is used to optimize generating policy to generate better candidates from the pair. For example, Reinforcement Learning from Human Feed-

back (RLHF) is a model-based algorithm to optimize preference learning (Ouyang et al., 2022). However, the RLHF process is complex, time-consuming, and unstable. The direct PO uses an off-policy algorithm to directly optimize the generating policy, eliminating the need for a reward model (Rafailov et al., 2023; An et al., 2023; Kang et al., 2023; Zhao et al., 2023). These approach are more data-efficient and stable. For example, DPO uses the Bradley-Terry model (Bradley and Terry, 1952) and log-loss, which can lead to over-fitting to the preference data, especially when preference is deterministic and ignores the KL-regularization term. The IPO algorithm (Azar et al., 2023) addresses this issue by using a root-finding MSE loss to solve the problem of ignoring KL-regularization when preference is deterministic. However, these methods fail to fully account for more fine-grained preferences and that is exactly what we want to do.

3 Preliminary

The main pipeline of preference learning usually consists of: 1) pretraining and Supervised Fine-Tuning (SFT), where SFT is not a must; 2) preference data collection; 3) preference optimization.

Pretraining and SFT Phase Preference learning typically starts with a pretrained LLMs or LLMs fine-tuned on high-quality data using maximum likelihood estimation. The final policy π_{ref} after this phase is represented as

$$\pi_{\text{ref}} \approx \arg\max_{\pi} \mathbb{E}_{x,y\sim\mathcal{D}_{\text{ref}}} \log \pi(x) \log(y|x), \quad (1)$$

where $\mathcal{D}_{\text{ref}}$ denotes the training data distribution.

Preference Data Collection Phase After pretraining and SFT phase, π_{ref} is prompted by context x, and generate two responses $y_w, y_l \sim \pi_{\text{ref}}(\cdot|x)$. Then x, y_w, y_l is labeled by humans to judge which response is preferred and denote $y_w \succ y_l|x$ if y_w is preferred, and $y_l \succ y_w|x$ if y_l is preferred. We define a new symbol $I = \mathbb{I}[y_w \succ y_l|x]$, and all $<x, y_w, y_l, I>$ consist the preference dataset $\mathcal{D}^p$:

$$\langle x, y_w, y_l, I\rangle \sim \mathcal{D}^p. \quad (2)$$

Preference Optimization Phase In the final phase, the prevailing method uses reinforcement learning algorithm to learn an explicit or implicit reward from the preference data, and then using on-policy or off-policy policy gradient algorithm to maximize the reward. Recently, some methods have derived the optimal policy using reward maximization under KL-regularization and also derive a loss with optimal policy as its solution, then learn the optimal policy by minimizing the derived loss on empirical dataset.

Reinforcement Learning from Human Feedback (RLHF) The RLHF uses standard two-phase reward model-based reinforcement learning to maximize the reward. It contains two steps: 1) reward estimation from preference data 2) reward maximization using PPO algorithm. It aims to maximize reward with a KL constraint on the reference model π_{ref} (inputs x omitted):

$$\pi^* = \arg\max_{\pi} \mathbb{E}_{y\sim\pi}\left[r(y) - \beta \log \frac{\pi(y)}{\pi_{\text{ref}}(y)}\right], \quad (3)$$

where β is the regularization weight and $r(y)$ is the reward function learned using the Bradley-Terry model on the preference dataset of generating y.

Direct Preference Optimization (DPO) DPO eliminates the training of reward model. It derives a loss on the current policy π_θ (y_w, y_l omitted):

$$\mathcal{L}_{dpo} = -\log\sigma\left(\beta \log \frac{\pi_\theta(y_w)}{\pi_{\text{ref}}(y_w)} - \beta \log \frac{\pi_\theta(y_l)}{\pi_{\text{ref}}(y_l)}\right), \quad (4)$$

i.e., the binary cross entropy with

$$\hat{p}_\theta(y_w \succ y_l) = \sigma\left(\beta \log \frac{\pi_\theta(y_w)}{\pi_{\text{ref}}(y_w)} - \beta \log \frac{\pi_\theta(y_l)}{\pi_{\text{ref}}(y_l)}\right), \quad (5)$$

and target $p(y_w \succ y_l) = 1$. We describe more PO methods in details in Appendix B.

4 Methodology

4.1 Problem Formulation

Formally, consider a query q and a corpus of text documents $\mathcal{D}$. The goal is to produce an output $\mathcal{S}$, where $\mathcal{S}$ is a collection of n distinct statements: $s_1, s_2, \ldots, s_n$. Each statement s_i is associated with a set of citations $\mathcal{C}_i$. This set $\mathcal{C}_i$ is defined as $\mathcal{C}_i = \{c_{i,1}, c_{i,2}, \ldots\}$, where each $c_{i,j}$ is a document from the corpus $\mathcal{D}$. For application purposes, the output from LLMs can be divided into individual statements using sentence boundaries. This approach is utilized because a single sentence typically encapsulates a coherent statement while maintaining brevity, facilitating easy verification. Regarding the citation format, citations are typically presented in square brackets, e.g., *The sun is formed approximately 4.6 billion years ago* `[1][2]`. However, it should be noted that these citations can

Dataset	Source	# Examples
Post-training		
EVIGSE	Internet	3508
ExpertQA	Internet	906
HAGRID	Wiki	1301+615(dev)
Preference Optimization		
stanford_alpaca	Wiki	7741
oasst1	Wiki	2478
asqa	Wiki	2333
sharegpt	Wiki	2490
wow	Wiki	3689
gpt4_alpaca	Wiki	6679
flan_v2	Wiki	1693
Test		
ASQA	Wiki	948
StrategyQA	Wiki	490
ELI5	Sphere	1000

Table 1: Statistics of data collections used at different stages in the APO framework.

be attributed to specific phrases as well, not just at the end of sentences.

Moreover, in this paper, we define *generation hallucination* refers to a situation where the model generates content that is not based on factual information and *attribution hallucination* means that the statement corresponding to one citation is unfaithful or not supported by the referred source content.

4.2 Overall Framework

As shown in Figure 2, we introduce the APO framework to apply preference learning for attribution task. The APO framework consists of the post-training procedure to ground the base model for attribution (§4.3), and the preference optimization procedure to address both generation hallucination and attribution hallucination (§4.4).

4.3 Post-training

The goal of post-training procedure is to ensure that given a specific question q and a corpus of text documents $\mathcal{D}$, the model can be successfully instructed to generate answer $\mathcal{S}$ and add citation $\mathcal{C}_i$ for each statement s_i in its response when necessary.

Data Collection We construct the post-training data from training sets using existing attribution datasets including EVIGSE (Liu et al., 2023), ExpertQA (Malaviya et al., 2023) and HARGID (Kamalloo et al., 2023). We select these datasets because they are high-quality attribution datasets with diverse domains and sources annotated by human experts or powerful LLMs. After preprocessing and formatting, the final post-training data collection includes 6,330 samples. The pre-processing details are shown in Appendices C and E, and the statistics of training data are shown in Table 1.

Algorithm 1 Preference data sampling and labeling

Input Queries Q, Critic M_c, Generator M_g, Retriever R
Output Output initialized preference dataset P_{init}
P_{init} = {}
for $q \in Q$ **do**
  Retrieve top-k passages D using R given q
  Predicts *relevant label* $\mathcal{L}_{rel} \in \{0, 1\}$ using critic M_c for each d in D given q, d
  Generate $\mathcal{S}$ constructed by statements $\{s_1, s_2, .., s_n\}$, using generator M_g given $(\mathcal{I}_{post}, q, D)$.
  for $s_i \in \mathcal{S}$ **do**
    Predicts *supported label* $\mathcal{L}_{sup} \in \{0, 1\}$ for each $c_{i,j}$ using critic M_c given q, $d \leftrightarrow c_{i,j}$, s_i
  end for
  Add augmented $(q, D, \mathcal{S}, \mathcal{C}, \mathcal{L}_{rel}, \mathcal{L}_{sup})$ to P_{init}
end for

Training After that, instruction $\mathcal{I}_{post}$, documents $\mathcal{D}$ and question q are formatted to be the input while answer $\mathcal{S}$ composed of multiple statements is formatted as output. We tune the model using autoregressive language modeling objectives, resulting in initial generator M_g.

4.4 Preference Optimization

In this section, we describe our preference optimization procedure to enable a model-agnostic approach for improving the quality of generated responses. First, considering the cost of labeling preference data, we devise an automatic data collection algorithm motivated by errors where previous models may have misattributed. Second, we propose a progressive preference optimization approach to amplify the preference signal by using synthesized preference pairs. We further apply the experience replay to alleviate the over-fitting and text degradation phenomenon due to the distribution shift introduced by automatic data generation.

Automatic Data Collection In general, attributed text generation should be both relevant and supported (Asai et al., 2023b). Being relevant needs the reference document in the answer to be helpful in handling the question. It is used to measure whether $\mathcal{C}$ provides useful information to solve q. Being supported asks the generated text be grounded on the reference documents. It is used to measure whether all of the verification-worthy statements in $\mathcal{S}$ are supported by $\mathcal{C}$.

Following the requirements above, we first get initial responses and related labels for each query with the Algorithm 1. The query comes from multiple open domain tasks or high-quality instruc-

tion data sets shown in Table 1. The source of retrieved documents is English Wikipedia. The retriever R we use here is `gtr-t5-large`[2]. The objective is to generate the attributed text with *relevant* and *supported* labels for related documents using the critic model M_c. Here, we use pre-trained `selfrag_llama2_7b`[3] as M_c in Asai et al. (2023b) because it can give fine-grained feedback using reflection tokens.

After that, we generate preference pairs using an automatic collection algorithm. Specifically, we determine whether the citations $\mathcal{C}_i$ of each statement s_i of query q are all related to it based on the relevant tags. If it is all relevant, we add the current statement and its preceding statements $\mathcal{S}_i$ to the set P_{tmp} for subsequent processing. For example, if s_2 meets the requirement, we add $\{s_1, s_2\}$ to P_{tmp}. The motivation here is that we want to select the statements that can answer the question based on the document as the initial set. Then, for each entry in P_{tmp}, we first retrieve another top-m ($m \gg k$) documents and filter them into 10 irrelevant documents D_{ir} scored by relevant logits predicted by M_c. If all documents in D_{ir} are relevant, we use the last 10 documents as D_{ir}. After that, we generate the positive and negative pair for each statement $s_i \in P_{tmp}$. There are two situations: the statement s_i is fully supported by $\mathcal{C}_i$ and otherwise. For the first situation, we first expand $\mathcal{C}_i$ with supported document by second judgment in D using M_c. Then, we generate one positive statement using q, $\mathcal{S}_{i-1}$ and new $\mathcal{C}_i$ and two negative statements using q, $\mathcal{S}_{i-1}$, D_{ir} and q, $\mathcal{S}_{i-1}$, new $\mathcal{C}_i$, error instruction e respectively. Thus, there are two preference pairs in this context. For the second situation, we generate one positive statement using q, $\mathcal{S}_{i-1}$ and new $\mathcal{C}_i$ and one negative statement using q, $\mathcal{S}_{i-1}$, $D - \mathcal{C}_i$, error instruction e. The full procedure is shown in Algorithm 2.

In the generation of negative samples, we use the error instruction $e \in \mathcal{E}$, which defines two types: *irrelevant but supported* means the generated text s_i is grounded on unhelpful reference documents $\mathcal{C}_i$, while *relevant but unsupported* further has three fine-grained subtypes: 1) *fabricated statement* refers to the generated text contains facts or information that cannot be derived from reference documents; 2) *mistaken synthesis* means that several reference documents are used, but facts or logics are mistakenly intermingled. The generated text thus contains factual error or logic error; 3) *unintentional omission* means that reference documents are used, but the key points are incomplete. There are no factual errors in generated text, but some information is omitted. The *irrelevant but supported* error derives from attribution hallucination, whereas the *relevant but unsupported* error is the result of generation hallucination. Note that irrelevant and unsupported errors are not included, since it is more like easy negatives. The details of error instructions are in Appendix F.

Progressive Preference Optimization To reinforce the preference feature and alleviate sparse reward problem (Zheng et al., 2023; Lightman et al., 2023), we propose a progressive preference optimization method. Considering generations can be separated into several consecutive statements, each statement may contain hallucinations at all. The entire response-level reward preference modeling performs in the global context and potentially oversights the fine-grained deterministic preferences we constructed. Hence, we use fine-grained statement-level reward to perform preference optimization to update the model in a more effective and efficient way. Formally, assuming that deterministic preference is performed at statement-level, we can rewrite the preference optimization loss in Eqn. (4) as follows ($-\log\sigma$ omitted):

$$
\begin{aligned}
\mathcal{L} &\triangleq \beta \log \frac{\pi_\theta(y_w)}{\pi_{ref}(y_w)} - \beta \log \frac{\pi_\theta(y_l)}{\pi_\theta(y_w)} \\
&= \beta \log \frac{\sum_i \pi_\theta(s_i^w | s_{:i-1}^w)}{\sum_i \pi_{ref}(s_i^w | s_{:i-1}^w)} - \beta \log \frac{\sum_j \pi_\theta(s_j^l | s_{:j-1}^l)}{\sum_j \pi_{ref}(s_j^l | s_{:j-1}^l)} \\
&= \beta \sum_i \log \frac{\pi_\theta(s_i^w | s_{:i-1}^w)}{\pi_{ref}(s_i^w | s_{:i-1}^w)} - \beta \sum_j \log \frac{\pi_\theta(s_j^l | s_{:j-1}^l)}{\pi_{ref}(s_j^l | s_{:j-1}^l)} \\
&= \beta \sum_i \left(\log \frac{\pi_\theta(s_i^w | s_{:i-1}^w)}{\pi_{ref}(s_i^w | s_{:i-1}^w)} - \log \frac{\pi_\theta(s_i^l | s_{:i-1}^l)}{\pi_{ref}(s_i^l | s_{:i-1}^l)} \right).
\end{aligned} \tag{6}
$$

The progressive preference optimization loss can be further written as follows ($-\log\sigma$ omitted):

$$
\begin{aligned}
\mathcal{L} &\triangleq \mathop{\mathbb{E}}_{(s_i^w, s_i^l \sim D)} \left(\beta \log \frac{\pi_\theta(s_i^w)}{\pi_{ref}(s_i^w)} - \beta \log \frac{\pi_\theta(s_i^l)}{\pi_{ref}(s_i^l)} \right) \\
&= \mathop{\mathbb{E}}_{(y_w, y_l \sim D)} \frac{1}{n} \sum_i \left(\beta \log \frac{\pi_\theta(s_i^w)}{\pi_{ref}(s_i^w)} - \beta \log \frac{\pi_\theta(s_i^l)}{\pi_{ref}(s_i^l)} \right).
\end{aligned} \tag{7}
$$

The main difference between vanilla preference optimization in Eqn. (4) and progressive preference optimization is that the latter contains an implicit mean pooling procedure when implementing the preference optimization loss.

[2] huggingface.co/sentence-transformers/gtr-t5-large

[3] huggingface.co/selfrag/selfrag_llama2_7b

Furthermore, the directed preference optimization may face the challenges of overfitting to some deterministic preference due to weak KL constraint (Azar et al., 2023). Hence, we propose to leverage experience replay (Rolnick et al., 2019) as learning with rehearsal to alleviate the over-fitting phenomenon. The idea of replaying experience typically stores a few old training samples within a small memory buffer. Therefore, we iteratively add post-training autoregressive language modeling loss to the preference optimization procedure in a fixed interval, resulting in final generator M_p.

4.5 Inference and Refinement

During inference, for query q, D is first retrieved and then sent to M_p output to the final answer $\mathcal{S}_{init}$ consists of n statements. As there may not be all statements correctly attributing documents, we additionally perform the post-hoc refinement after the original generation. We maintain a collection of citations $\mathcal{C}_{tmp}$. Starting from the last statement of $\mathcal{S}_{init}$, if the current s_i has the citations, update the $\mathcal{C}_{tmp}$ to the citations of the current s_i; if the current s_i does not have a citation, add the current citation set $\mathcal{C}_{tmp}$ to this statement until all n statements have been traversed. Then we concatenate these n statements together as the final answer $\mathcal{S}$.

5 Setup

5.1 Datasets and Evaluation Metrics

Dataset We mainly focus on attributable long-form question-answering (QA) task using ASQA dataset and ELI5 subsets from Gao et al. (2023b). In addition to these factoid long-form QA tasks, we test the generation quality on StrategyQA dataset (Geva et al., 2021) which focuses on open-domain QA where the required reasoning steps are implicit in the question. We use the official test set as our evaluation set.

Metrics Following Gao et al. (2023b), we report citation **recall**, **precision**, and **F1** which uses TRUE (Honovich et al., 2022) as the attribution evaluation model ϕ to automatically examine whether the cited documents entail the model generation. For ASQA dataset, we report the **recall of correct short answers** (EM-R) by checking whether the short answers (provided by the dataset) are exact substrings of the generation. For ELI5 dataset, we report the **claim recall** (Claim) to check whether the model output entails the sub-claims, that are generated by `text-davinci-003` (Ouyang et al., 2022). For StrategyQA dataset, we report the **accuracy** for task performance.

5.2 Competitive Methods

We compare APO with several baselines. For each baseline, we use `gtr-t5-large` as our retriever.

In-Context Learning (ICLCITE): We prompt LLMs with few-shot examples, each consisting of a query, a set of retrieved documents and an answer with inline citations. The LLMs can in-context learn from the examples and generate grounded responses for the test query and retrieved documents.

Post-Hoc Cite (POSTCITE): Given query q, we first instruct LLMs to answer q *without* retrieved documents. Then, we use the attribution evaluation model ϕ to link each statement to the most relevant document retrieved by the query.

Post-Hoc Attribute (POSTATTR): Instead of citing the most relevant document, for each statement, we further retrieve a set of *k* documents and then use the ϕ to link to the document that maximally supports the statement by threshold.

Self-RAG (Asai et al., 2023b): Self-RAG is the state-of-the-art (SoTA) method that adaptively retrieves documents on-demand. It generates with reflection on retrieved documents and its generations by special token control.

AGREE (Ye et al., 2023b): AGREE leverages test-time adaptation to reinforce unverified statements which iteratively improves the responses of LLMs. It tunes a pre-trained LLM to self-ground its response in retrieved documents using automatically collected data.

5.3 Implementation Details

If not specified, we retrieve the top 5 documents as the related documents to q and we set the decoding temperature to 0.01 during inference. For the post-training, we tune the model for 2 epochs with a learning rate of 5e-5. For the preference optimization, we tune the model with LoRA (Hu et al., 2022) for 1 epoch, and we set alpha to 2 and lora ranks to 16. We set m to 100. We use `llama-2-13b-base` (Touvron et al., 2023) for fair comparison. We run all the experiments on NVIDIA A100 80G GPUs.

6 Results

6.1 Main Result

Table 2 shows the comparison results of APO with other baselines on three datasets. In terms of cor-

Dataset & Metrics	ASQA				StrategyQA				ELI5			
	Correct	Citation			Correct	Citation			Correct	Citation		
	EM-R	Rec	Prec	F1	ACC	Rec	Prec	F1	Claim	Rec	Prec	F1
ICLCITE (Gao et al., 2023b)	35.2	38.4	39.4	38.9	**65.5**	20.6	33.1	25.4	13.4	17.3	15.8	16.5
POSTCITE (Gao et al., 2023b)	25.0	23.6	23.6	23.6	64.3	8.7	8.7	8.7	7.1	5.7	5.8	5.8
POSTATTR (Ye et al., 2023b)	25.0	33.6	33.6	33.6	64.3	12.5	12.5	12.5	7.1	12.2	12.2	12.2
Self-RAG (Asai et al., 2023b)	31.7	70.3	**71.3**	70.8	62.1	31.4	36.5	33.8	10.7	20.8	22.5	21.6
AGREE (Ye et al., 2023b)	39.4	64.0	66.8	65.4	64.6	30.2	37.2	33.3	9.4	21.6	16.0	18.4
APO (only post-training)	36.6	65.0	62.1	63.5	62.5	30.7	30.1	30.4	13.0	18.5	17.9	18.2
APO (our method)	**40.5**	**72.8**	69.6	**71.2**	61.8	**40.0**	**39.1**	**39.6**	**13.5**	**26.0**	**24.5**	**25.2**

Table 2: The performance comparison between our method and extensive baselines. Experiments are evaluated on ASQA (Stelmakh et al., 2022), StrategyQA (Geva et al., 2021) and ELI5 dataset (Fan et al., 2019). For most baselines, we use the results of previous works (Gao et al., 2023b; Ye et al., 2023b).

Method	EM-R	Rec	Prec	F1
Our Method	36.6	65.0	62.1	63.5
w/o asqa	38.8	71.7	67.2	69.4
w/o hallucinated statement	40.4	69.3	65.3	67.3
w/o mistaken synthesis	40.2	**73.4**	69.2	71.2
w/o unintentional omission	39.1	72.7	68.2	70.4
w/ response-level PO	38.9	69.1	65.1	67.1
w/ statement-level PO	**40.5**	72.8	**69.6**	**71.2**

Table 3: Ablation study on the ASQA dataset. We ablate not only the source and predefined error type used to construct PO data, but also the training strategy.

rectness and citation quality, our method outperforms the baselines on all three datasets. It shows that APO has better overall generation performance in various scenarios. Specifically, our method outperforms Self-RAG by 8.8 points on the EM-R metric. We speculate that this inconsistency stems from the difference between coherent generation and step-wise generation in Self-RAG. Our method also shows consistent improvements over AGREE across multiple benchmarks which suggests that APO can more effectively exploit the power of LLM to enhance retrieval. APO can be used to complement these active or adaptive retrieval-based methods and we leave it for future work. Compared to the post-training baseline, the preference optimization shows further improvement with an 8.0 average increased citation F1. Furthermore, we observe a trade-off between correctness and citation quality in several baselines including Self-RAG and AGREE, possibly due to the generation hallucination and attribution hallucination defined in §4.1. In contrast, APO helps to deal with these hallucinations and performs well in terms of both correctness and citation quality.

6.2 Ablation Study

We evaluate the effectiveness of each predefined error type and the results are shown in Table 3. Specifically, we perform progressive PO on the model after post-training and remove data corresponding to a predefined type. We observe that without data corresponding to hallucinated statement error, citation F1 drops significantly which suggests that our approach improves the groundedness of the model. Mistaken synthesis error seems to contribute little to performance improvement, but we observe that it can help improve groundedness under human evaluation (§6.5). Without unintentional omission error, the model shows poor generation quality. This means that the model may generate incomplete answers.

Moreover, we perform an ablation study on the training strategy of preference optimization. We find that the model can also be improved under the response-level preference optimization method such as vanilla DPO, but the improvement is slightly less. In addition, we ablate the PO by removing the ASQA questions from our preference data. Note that we construct the preference data based on the training set of ASQA, and use its test set for evaluation. We have verified and guaranteed that there is no data overlap between the two. We find that the generation quality and citation quality have decreased. We attribute it to high-quality in-domain questions in ASQA as a long-form question answering dataset.

6.3 Different Prompting Strategy

We explore applying APO to four prompting strategies (Gao et al., 2023b): 1) VANILLA that provides the top-5 retrieved documents for each question. It is our default setting. 2) SUMM that provides summaries instead of the full text of the top-10

Method & Metrics	ASQA			
	Correct	Rec	Prec	F1
llama-2-13b-chat				
VANILLA(5-psg)	32.6	60.0	52.1	55.8
SUMM(10-psg)	42.9	58.7	50.4	54.2
SNIPPET(10-psg)	41.3	57.4	52.1	54.6
ORACLE(5-psg)	41.4	54.5	52.9	53.7
Our method				
VANILLA(5-psg)	40.5	**72.8**	**69.6**	**71.2**
SUMM(10-psg)	42.7	60.9	53.4	56.9
SNIPPET(10-psg)	42.3	57.8	51.6	54.5
ORACLE(5-psg)	**52.4**	70.5	66.2	68.3
Method & Metrics	**ELI5**			
	Correct	**Rec**	**Prec**	**F1**
llama-2-13b-chat				
VANILLA(5-psg)	12.1	16.4	19.7	17.9
SUMM(10-psg)	6.1	9.9	14.3	11.7
SNIPPET(10-psg)	11.9	29.4	28.6	29.0
ORACLE(5-psg)	16.9	21.4	27.3	24.0
Our method				
VANILLA(5-psg)	13.5	26.0	24.5	25.2
SUMM(10-psg)	12.7	**37.8**	**35.7**	**36.7**
SNIPPET(10-psg)	14.2	37.6	34.8	36.1
ORACLE(5-psg)	**21.7**	32.6	30.8	31.7

Table 4: Comparisons with different retrieval context.

Method & Metrics	ASQA			
	Correct	Rec	Prec	F1
APO (only post-training)	36.6	65.0	62.1	63.5
w/ Positive statement SFT	29.0	66.7	56.8	61.4
w/ IPO (Azar et al., 2023)	39.9	72.7	69.2	70.9
w/ SLiC (Zhao et al., 2023)	40.1	72.5	69.1	70.8
w/ KTO (Kawin et al., 2023)	39.8	72.5	68.7	70.5
w/ Progressive PO	**40.5**	**72.8**	**69.6**	**71.2**
Method & Metrics	**ELI5**			
	Correct	**Rec**	**Prec**	**F1**
APO (only post-training)	13.0	18.5	17.9	18.2
w/ Positive statement SFT	10.6	**34.5**	**30.8**	**32.5**
w/ IPO (Azar et al., 2023)	13.5	26.5	24.8	25.6
w/ SLiC (Zhao et al., 2023)	13.7	30.7	22.0	25.6
w/ KTO (Kawin et al., 2023)	**14.3**	24.7	26.5	25.6
w/ Progressive PO	13.5	26.0	24.5	25.2

Table 5: Comparisons with different preference method.

Error Type	# Proportion (%)
Attribution hallucination	26.4
Generation hallucination	
- Fabrication	48.4
- Omission	18.7
- Synthesis	6.5

Table 6: Error types of the proposed methods.

retrieved documents for each question. 3) SNIPPET that provides snippets instead of the full text of the top 10 retrieved documents for each question. 4) ORACLE that provides 5 gold documents for each question. We use `llama-2-13b-chat` as the comparison method because it has impressive instruction following ability and moderate size. As shown in Table 4, we find that in most cases, APO achieves better performance than baseline. For example, APO under VANILLA and ORACLE settings performs best in Citation F1 on ASQA, while it under SUMM and SNIPPET settings in ELI5 has improved Citation F1. It shows that the format of the context has an impact on attribution task.

6.4 Different PO Methods

Table 5 illustrates the results of different direct preference optimization methods adopted by M_p. We include a SFT baseline to tune the M_g using the positive part in the chosen preference pairs that we created. We observe that our method can be transferred to several different preference optimization methods, but the performance swings in several metrics. All preference optimization methods have performance boosts compared with the post-training baseline and the SFT baseline. It shows that preference optimization can help improve the generation quality to some extent.

6.5 Error Analysis

We conduct human evaluation of model response on ASQA dataset. Specifically, we collect 50 samples that contain errors judged by the attribution evaluation model ϕ. We then perform a detailed manual review of these samples to identify error types. Our evaluation results are shown in Table 6. We find that nearly half of the errors are of fabrication error. We reveal that the model either generated text not supported by the reference documents or incorrectly attributed information to irrelevant documents. In certain instances, hallucinations are due to the documents with low quality. For example, some documents are truncated, and the model attempts to complete or extrapolate the incomplete text. Additionally, we notice omission errors on both generated text and citation where the model fails to generate necessary citations to substantiate its statements. Although synthesis errors are less common, we observe some cases which model conflated information from multiple documents and generated counterfactual statements. The case study is shown in Appendix G.

7 Conclusion

This paper introduces the APO framework for attributed text generation. We treat attribution as a preference learning task, utilizing curated post-training collections and an automated synthesis algorithm to reduce manual labeling costs. Experiments on three datasets demonstrate the effectiveness of APO which achieves leading citation F1 and improved response quality. Future work can explore extending APO to real-world applications.

Limitation

We aim to improve the credibility and reliability of content generated by LLMs using the APO framework. However, it faces limitations such as the narrow scope of datasets used, which may not fully represent the diversity of real-world applications (Liu et al., 2023). The generalization capabilities of the model are also a concern, as the automatic generated data may not cover all scenarios of hallucination. While addressing the high cost of data labeling, the scalability and economic feasibility in larger datasets remain unexplored. The approximation of human citation processes may not capture all the complexities of scholarly writing, and its reliance on external sources raises concerns about the quality and availability of these sources. Potential biases in training data and synthesized data could lead to biased outputs (Wang et al., 2023; Hu et al., 2015). The robustness of the framework against deliberate hallucination and its adaptability to rapidly evolving NLP fields are not fully assessed, highlighting areas for future improvement and research in enhancing LLM reliability.

Ethical Statement

The ethical considerations surrounding the use of LLMs that generate citations encompass a range of concerns, including the risk of increased trust without verification, challenges in time-critical decision-making, the assumption of inherent trustworthiness, and copyright issues. Ethically, it is crucial to encourage users to critically engage with and verify machine-generated content to mitigate misinformation and hallucinations. Additionally, recognizing the limitations and potential legal challenges related to copyright when using such attributions is essential. Addressing these ethical issues and educating users on the potential pitfalls becomes increasingly important to ensure the responsible and informed use of text generated by LLMs.

Acknowledgments

We thank the anonymous reviewers for their comments and suggestions. This work is supported by grants: Natural Science Foundation of China (No. 62376067) and Guangdong Basic and Applied Basic Research Foundation (2023A1515110078). Xuebo Liu was sponsored by CCF-Tencent Rhino-Bird Open Research Fund.

References

Gaon An, Junhyeok Lee, Xingdong Zuo, Norio Kosaka, Kyung-Min Kim, and Hyun Oh Song. 2023. Direct preference-based policy optimization without reward modeling. In *Neural Information Processing Systems*.

Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Millican, David Silver, Slav Petrov, Melvin Johnson, Ioannis Antonoglou, Julian Schrittwieser, Amelia Glaese, Jilin Chen, Emily Pitler, Timothy P. Lillicrap, Angeliki Lazaridou, Orhan Firat, James Molloy, Michael Isard, Paul Ronald Barham, Tom Hennigan, Benjamin Lee, Fabio Viola, Malcolm Reynolds, Yuanzhong Xu, Ryan Doherty, Eli Collins, Clemens Meyer, Eliza Rutherford, Erica Moreira, Kareem Ayoub, Megha Goel, George Tucker, Enrique Piqueras, Maxim Krikun, Iain Barr, Nikolay Savinov, Ivo Danihelka, Becca Roelofs, Anaïs White, Anders Andreassen, Tamara von Glehn, Lakshman Yagati, Mehran Kazemi, Lucas Gonzalez, Misha Khalman, Jakub Sygnowski, and et al. 2023. Gemini: A family of highly capable multimodal models. *ArXiv preprint*, abs/2312.11805.

Akari Asai, Sewon Min, Zexuan Zhong, and Danqi Chen. 2023a. Retrieval-based language models and applications. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts, ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 41–46.

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

Mohammad Gheshlaghi Azar, Mark Rowland, Bilal Piot, Daniel Guo, Daniele Calandriello, Michal Valko, and Rémi Munos. 2023. A general theoretical paradigm to understand learning from human preferences. *ArXiv preprint*, abs/2310.12036.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan,

Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom B. Brown, Jack Clark, Sam McCandlish, Chris Olah, Benjamin Mann, and Jared Kaplan. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *ArXiv preprint*, abs/2204.05862.

Bernd Bohnet, Vinh Q. Tran, Pat Verga, Roee Aharoni, Daniel Andor, Livio Baldini Soares, Jacob Eisenstein, Kuzman Ganchev, Jonathan Herzig, Kai Hui, Tom Kwiatkowski, Ji Ma, Jianmo Ni, Tal Schuster, William W. Cohen, Michael Collins, Dipanjan Das, Donald Metzler, Slav Petrov, and Kellie Webster. 2022. Attributed question answering: Evaluation and modeling for attributed large language models. *ArXiv preprint*, abs/2212.08037.

Ralph Allan Bradley and Milton E Terry. 1952. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345.

Terrence A Brooks. 1986. Evidence of complex citer motivations. *Journal of the American Society for Information Science*, 37(1):34–36.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Paul F. Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 4299–4307.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. 2023. A survey for in-context learning. *ArXiv preprint*, abs/2301.00234.

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.

Luyu Gao, Zhuyun Dai, Panupong Pasupat, Anthony Chen, Arun Tejasvi Chaganty, Yicheng Fan, Vincent Y. 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), ACL 2023, Toronto, Canada, July 9-14, 2023*, 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, EMNLP 2023, Singapore, December 6-10, 2023*, pages 6465–6488.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, and Haofen Wang. 2023c. Retrieval-augmented generation for large language models: A survey. *ArXiv preprint*, abs/2312.10997.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361.

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 Second DialDoc Workshop on Document-grounded Dialogue and Conversational Question Answering*, pages 161–175.

Baotian Hu, Qingcai Chen, and Fangze Zhu. 2015. LCSTS: A large scale chinese short text summarization dataset. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 1967–1972. The Association for Computational Linguistics.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*.

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*, abs/2307.16883.

Yachen Kang, Diyuan Shi, Jinxin Liu, Li He, and Donglin Wang. 2023. Beyond reward: Offline preference-guided policy optimization. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume

202 of *Proceedings of Machine Learning Research*, pages 15753–15768.

Ethayarajh Kawin, Xu Winnie, Jurafsky Dan, and Kiela Douwe. 2023. Human-centered loss functions (halos). Technical report, Contextual AI.

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*, abs/2311.03731.

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

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

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. *ArXiv preprint*, abs/2305.20050.

Nelson F. Liu, Tianyi Zhang, and Percy Liang. 2023. Evaluating verifiability in generative search engines. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 7001–7025.

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

OpenAI. 2022. Chatgpt: Optimizing language models for dialogue.

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

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*, abs/2305.18290.

Hannah Rashkin, Vitaly Nikolaev, Matthew Lamm, Michael Collins, Dipanjan Das, Slav Petrov, Gaurav Singh Tomar, Iulia Turc, and David Reitter. 2021. Measuring attribution in natural language generation models. *ArXiv preprint*, abs/2112.12870.

David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy P. Lillicrap, and Gregory Wayne. 2019. Experience replay for continual learning. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 348–358.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *ArXiv preprint*, abs/1707.06347.

Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. 2021. Retrieval augmentation reduces hallucination in conversation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3784–3803.

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.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F. Christiano. 2020. Learning to summarize with human feedback. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

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*, abs/2312.09075.

Misha Teplitskiy, Eamon Duede, Michael Menietti, and Karim R Lakhani. 2022. How status of research papers affects the way they are read and cited. *Research Policy*, 51(4):104484.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas

Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *ArXiv preprint*, abs/2307.09288.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 13484–13508.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022. Emergent abilities of large language models. *Trans. Mach. Learn. Res.*, 2022.

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. 2020. Neural text generation with unlikelihood training. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*.

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*, abs/2304.14732.

Hongbin Ye, Tong Liu, Aijia Zhang, Wei Hua, and Weiqiang Jia. 2023a. Cognitive mirage: A review of hallucinations in large language models. *ArXiv preprint*, abs/2309.06794.

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

Xiang Yue, Boshi Wang, Kai Zhang, Ziru Chen, Yu Su, and Huan Sun. 2023. Automatic evaluation of attribution by large language models. *ArXiv preprint*, abs/2305.06311.

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

Yao Zhao, Rishabh Joshi, Tianqi Liu, Misha Khalman, Mohammad Saleh, and Peter J. Liu. 2023. Slic-hf: Sequence likelihood calibration with human feedback. *ArXiv preprint*, abs/2305.10425.

Rui Zheng, Shihan Dou, Songyang Gao, Yuan Hua, Wei Shen, Binghai Wang, Yan Liu, Senjie Jin, Qin Liu, Yuhao Zhou, Limao Xiong, Lu Chen, Zhiheng Xi, Nuo Xu, Wenbin Lai, Minghao Zhu, Cheng Chang, Zhangyue Yin, Rongxiang Weng, Wensen Cheng, Haoran Huang, Tianxiang Sun, Hang Yan, Tao Gui, Qi Zhang, Xipeng Qiu, and Xuanjing Huang. 2023. Secrets of RLHF in large language models part I: PPO. *ArXiv preprint*, abs/2307.04964.

Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul F. Christiano, and Geoffrey Irving. 2019. Fine-tuning language models from human preferences. *ArXiv preprint*, abs/1909.08593.

Guido Zuccon, Bevan Koopman, and Razia Shaik. 2023. Chatgpt hallucinates when attributing answers.

A Further experiments

We preform the results on ASQA and ELI5 datasets with more metrics across 3 seeds in Table7.

B Details about Preference Optimization Methods

Reinforcement Learning form Human Feedback (RLHF) (Ouyang et al., 2022) uses reward-model-based reinforcement learning algorithm to learn the optimal policy. It first learns a reward model from the preference data, then uses an on-policy PPO algorithm (Schulman et al., 2017) to maximize the learned reward. The reward is learned to use Bradley-Terry model (Bradley and Terry, 1952), which assumes the preference score can be approximated by substituted with point-wise reward. This assumption may lead to an approximation error when preference is deterministic. The PPO algorithm is used on data sampled from generating policy, which may have a different support or distribution drift from preference data, the learned reward model inference on the out-of-distribution data may reduce the accuracy. The process of RLHF needs to train reward model and on-policy PPO algorithm which is complex, time-consuming, and unstable.

Direct Preference Optimization (DPO) (Rafailov et al., 2023) combines off-policy algorithm and Bradley-Terry model to directly learn the generating policy from preference data. The off-policy algorithm is based on KL-regularization reward maximization from off-RL community, which is data efficient, stable and eliminating the need for a reward model. When preference is deterministic which occurs in most cases, the reward of Bradley-Terry model is undefined, which leads to ignoring the KL-regularization term and overfitting the preference dataset.

Identity-mapping Preference Optimization (IPO) (Azar et al., 2023) claims when preferences are deterministic or near deterministic, DPO will lead over-fitting to the preference dataset at the expense of ignoring the KL-regularation term. To optimize the objective, IPO derives an off-policy loss on empirical dataset:

$$h^{\pi_\theta}_{y_w,y_l} = \log \frac{\pi_\theta(y_w)}{\pi_{\text{ref}}(y_w)} - \log \frac{\pi_\theta(y_l)}{\pi_{\text{ref}}(y_l)}, \quad (8)$$

$$L_{\text{IPO}}(\theta, y_w, y_l) = \left(h^{\pi}_{y_w,y_l} - \frac{1}{2\beta} \right)^2. \quad (9)$$

That means IPO loss will always regularize π_θ towards π_{ref} by controlling the gap between the log-likelihood ratios $\log \frac{\pi_\theta(y_w|x)}{\pi_\theta(y_l|x)}$ and $\log \frac{\pi_{\text{ref}}(y_w|x)}{\pi_{\text{ref}}(y_l|x)}$.

Kahneman-Tversky Optimization (KTO) (Kawin et al., 2023) directly maximizes the utility of LLM generations instead of maximizing the log-likelihood of preferences by introducing a Kahneman-Tversky Optimization loss. KTO does not need preference pairs and only knowledge of whether output is desirable or undesirable for a given input.

Sequence Likelihood Calibration (SLiC) (Zhao et al., 2023) uses calibrated likelihood of model-generated sequences to better align with reference sequences in the model's latent space. It tries to alleviate the problem of MLE that gives probability mass to sparsely observed target sequences, which is used to calculate reward in DPO.

C Details about Pre-processing

For ExpertQA dataset, we remove samples whose 1) citations attribute to empty references; 2) documents contain different document IDs but same context. For EVIGSE dataset, we remove samples whose 1) citation attribute to "None" references; 2) do not have reference documents. We further normalize the "supported" label and the citation format for these datasets. The details of each dataset we used for post-training procedure after pre-processing are shown in Table 8.

D Details about Preference Data Creation

For each error type, we set the weight to be 1:1:1 to make sure the error samples are balanced. The quantitative assessment about preference dataset is shown in Table 9.

E Post-training Templates

The post-training template we used follows the question answering template used by Gao et al. (2023b) since we find that preposition question before document can result in a performance boost when trying ICLCITE method in the preliminary experiments. The concrete templates are shown in Table 10.

F Details about the Instruction

The templates employed for generating preference data are detailed in Table 11 for positive instances,

Method & Metrics	ASQA				
	EM	**Rec**	**Prec**	**MAUVE**	**rougeLSUM**
APO (only post-training)	36.59(0.12)	65.57(0.17)	62.01(0.23)	77.68(0.53)	37.68(0.02)
APO (only preference optimization)	35.10(0.31)	50.57(1.74)	38.02(2.17)	74.68(0.40)	35.92(0.72)
APO (our method)	40.43(0.02)	73.01(0.16)	69.62(0.18)	73.24(1.74)	37.10(0.02)
Method & Metrics	**ELI5**				
	Correct	**Rec**	**Prec**	**MAUVE**	**rougeLSUM**
APO (only post-training)	13.04(0.03)	18.47(0.04)	17.84(0.06)	34.84(2.53)	21.19(0.17)
APO (our method)	13.53(0.17)	26.40(0.09)	24.91(0.06)	24.91(0.06)	20.79(0.06)

Table 7: Experiment results on ASQA and ELI5 with more metrics across 3 seeds.

Dataset	# Sample	Avg. Query Length	Avg. Response Length	Avg. Statements	Avg. Citations
EVIGSE	3508	51.03	379.05	4.32	3.19
ExpertQA	906	106.90	999.84	7.16	5.67
HARGID(train)	1301	38.55	368.22	4.62	2.85
HARGID(dev)	615	40.43	292.46	3.63	2.54

Table 8: Details for our post-training data after pre-processing.

Table 12 for statements exhibiting hallucination errors, Table 13 for statements with synthesis errors, and Table 14 for statements characterized by omission errors.

G Case Study

In this section, we perform a detailed case study and demonstrate several examples of each type of error we defined. As shown in Table 15, we classify it as a fabrication error since it uses an undefined entity. In Table 16, we classify it as a synthesis error since it mixes up facts from document 4 and document 5, which results in a factual error. In Table 17, we classify it as a omission error since it used facts from document 4 and document 5, but document 4 is not attributed.

H Related Works of Retrieval Augmentation of LLMs

Retrieval augmentation has emerged as a prominent technique aimed at enhancing the accuracy and veracity of LLMs (Gao et al., 2023c; Asai et al., 2023a). Specifically, Sun et al. (2023) couples LLMs with long-term and short-term memories, resulting in improved claim and citation generation. Meanwhile, in order to effectively incorporate external knowledge into LLMs, SearChain proposes a global reasoning chain strategy that facilitates retrieval augmentation generation at each node within the chain (Xu et al., 2023). In another line of research, the self-reflection is leveraged for retrieval verification during the retrieval-augmented generation process (Li et al., 2023b; Asai et al., 2023b). Despite these advancements, prior studies have not adequately addressed the issue of attribution hallucination (Zuccon et al., 2023). In contrast, we focus on making the model better answer the query and align with the reference.

Type	SubType	num	Avg. Citations	Avg. Length
Supported & Relevant	-	95,623	1.77	173.86
Supported & Irrelevant	-	69,825	1.98	148.02
Unsupported & Relevant	Fabrication	31,600	1.77	110.42
Unsupported & Relevant	Omission	21,202	1.78	93.59
Unsupported & Relevant	Synthesis	42,821	1.78	212.19

Table 9: Details for our preference data.

Algorithm 2 Automatic preference data collection algorithm

Input Input P_{init}, Critic M_c, Generator M_g, Retriever R, Error instructions $\mathcal{E}$
Output Output augmented preference dataset P_{syn}
P_{tmp} = { }
for $(q, D, \mathcal{S}, \mathcal{C}, \mathcal{L}_{rel}, \mathcal{L}_{sup})$ in P_{init} **do**
    for each statement s_i in $\mathcal{S}$ **do**
        if the referenced passages $\mathcal{C}_i$ are all *relevant* to q **then**
            add $(q, D, \mathcal{S}_{:i}, \mathcal{C}_{:i}, \mathcal{L}_{rel}, \mathcal{L}_{sup}^{:i,:})$ to P_{tmp}
        end if
    end for
end for
for $(q, D, \mathcal{S}_{:i}, \mathcal{C}_{:i}, \mathcal{L}_{rel}, \mathcal{L}_{sup}^{:i,:})$in P_{tmp} **do**
    Retrieve top-m passages D_{ir} using retriever R given q.
    Subsequently delete passage d from D_{ir}, if d_i is predicted as *relevant* to q using critic M_c.
    for each statement s_i in $\mathcal{S}$ and its relative attributed passages $\mathcal{C}_i$ **do**
        if s_i is supported by $\mathcal{C}_i$ **then**
            Predicts *supported* for s_i using critic M_c given q, d_i, s_i, where $d_i \in D$
            Add *supported* passages to the relative attributed passages $\mathcal{C}_i$ of statement s_i.
            Generate $s_i^{s\wedge r}$ using M_g given q, $s_{:i-1}$, new $\mathcal{C}_i$.
            Generate $s_i^{s\wedge \tilde{r}}$ using M_g given q,$s_{:i-1}$, D_{ir}.
            Generate $s_i^{\tilde{s}\wedge r}$ using M_g given q, $s_{:i-1}$, new $\mathcal{C}_i$ and pre-defined error type e.
            add $(q, s_i^{s\wedge r}, s_i^{\tilde{s}\wedge r}, D)$ and $(q, s_i^{s\wedge r}, s_i^{s\wedge \tilde{r}}, D)$ to P_{syn}.
        else
            Generate $s_i^{s\wedge r}$ using M_g given q, $s_{:i-1}$, new $\mathcal{C}_i$.
            Generate $s_i^{\tilde{s}\wedge r}$ using M_g given q, $s_{:i-1}$, $D-$ new $\mathcal{C}_i$ and pre-defined error type e.
            add $(q, s_i^{s\wedge r}, s_i^{\tilde{s}\wedge r}, D)$ to P_{syn}.
        end if
    end for
end for

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

Output
{{output}}

Table 10: Post-training Template with instruction $\mathcal{I}_{post}$

Input
Task: Your job is to write a high quality response with requirements as follows:
General: Given Request, incomplete response and evidence, continue write a single sentence as the next sentence of the unfinished response. If text in unfinished response is "None", you should start the response(the first sentence).
Detail: You should always use the facts from the evidences to propuse your response. Your response is correct and comprehensive, fully supported by the evidence we provided. **Don't use any evidence that can be directly retrieved from the evidences we provided**. No hallucinations, no factual errors, no logic errors.

Request: {{request}}
Evidence:
Document [1](Title: {{title 1}}): {{context 1}}
Document [2](Title: {{title 2}}): {{context 2}}
Document [3](Title: {{title 3}}): {{context 3}}
...
Document [n](Title: {{title n}}): {{context n}}
Unfinished response: {{past statements}}
Next sentence(good):

Output
{{output}}

Table 11: Positive Template

Input
Task: Your job is to write a low quality response with requirements as follows:
General: Given Request, incomplete response and evidence, continue write a single sentence as the next sentence of the unfinished response. If text in unfinished response is "None", you should start the response(the first sentence).
Detail: You will always ignore the evidence. On one hand, you won't follow the evidence we provided, your response should be irrelevant to the evidence we provided. On the other hand, your response should be relevant to the unfinished response.

Request: {{request}}
Evidence:
Document [1](Title: {{title 1}}): {{context 1}}
Document [2](Title: {{title 2}}): {{context 2}}
Document [3](Title: {{title 3}}): {{context 3}}
...
Document [n](Title: {{title n}}): {{context n}}
Unfinished response: {{past statements}}
Raw sentence(good): {{positive statement}}
Worse sentence(bad, ignore the evidence):

Output
{{output}}

Table 12: Negative, fabrication template

Input
Task: Your job is to write a low quality response with requirements as follows:
General: Given Request, incomplete response and evidence, continue write a single sentence as the next sentence of the unfinished response. If text in unfinished response is "None", you should start the response(the first sentence).
Detail: You should first, identify the relationships and entities in evidence; second, continue writing the next sentence of the response span with regard to the evidence. In your response, the relationships and entities should be mistakenly intermingled(you are making negative samples, we need low-quality data).

Request: {{request}}
Evidence:
Document [1](Title: {{title 1}}): {{context 1}}
Document [2](Title: {{title 2}}): {{context 2}}
Document [3](Title: {{title 3}}): {{context 3}}
...
Document [n](Title: {{title n}}): {{context n}}
Unfinished response: {{past statements}}
Raw sentence(good): {{positive statement}}
Worse sentence(bad, entities in evidences mistakenly intermingled):

Output
{{output}}

Table 13: Negative, synthesis template

Input
Task: Your job is to write a low quality response with requirements as follows:
General: Given Request, unfinished response and next sentence, omit some important points from the next sentence(good) and convert it into a worse response. Your converted worse response should be consistent with the unfinished response.

Request: List the ingredients needed to make a peanut butter and jelly sandwich
Unfinished response:
Raw sentence(good): To make a peanut butter and jelly sandwich, you will need peanut butter, jelly or jam of your choice, and bread.
Worse sentence(bad, omit the evidence): To make a peanut butter and jelly sandwich, you will need peanut butter and bread.

Request: What are the three features of a cloud-based Database-as-a-Service (DBaaS)?
Unfinished response: The three main features of a cloud-based DBaaS are scalability, cost efficiency, and backups. Scalability allows you to increase or decrease the resources used by the DBaaS with ease.
Raw sentence(good): Cost efficiency is another important feature of a cloud-based DBaaS, as it allows you to pay for only the resources you need and eliminates the need for upfront hardware investments.
Worse sentence(bad, omit the evidence): Cost efficiency is another important feature of a cloud-based DBaaS, as it allows you to pay for only the resources you need.

Request: {{request}}
Unfinished response: {{past statements}}
Raw sentence(good): {{positive statement}}
Worse sentence(bad, omit the evidence):

Output
{{output}}

Table 14: Negative, omission template

Question
When did the rams go to st louis?
Documents
Document [1](Title: History of St. Louis): 2011, with performances by Jay Leno and Aretha Franklin. In January 1995, Georgia Frontiere, the owner of the National Football League team known as the Los Angeles Rams (now St. Louis Rams), announced she would move that team to St. Louis. The team replaced the St. Louis Cardinals (now Arizona Cardinals), an NFL franchise that had moved to St. Louis in 1960 but departed for Arizona in 1988. The Rams played their first game in their St. Louis stadium, the Edward Jones Dome, on October 22, 1996. Starting in the early 1980s, more rehabilitation and construction projects began, some of
Document [2](Title: History of the St. Louis Rams): History of the St. Louis Rams The professional American football franchise now known as the Los Angeles Rams played in St. Louis, Missouri, as the St. Louis Rams from the 1995 through the 2015 seasons before relocating back to Los Angeles where the team had played from the 1946 season to the 1994 season. The Rams franchise relocated from Los Angeles to St. Louis in 1995, which had been without a National Football League (NFL) team since the Cardinals moved to Phoenix, Arizona in 1988. The Rams' first home game in St. Louis was at Busch Memorial Stadium against the
Document [3](Title: History of the St. Louis Rams): History of the St. Louis Rams The professional American football franchise now known as the Los Angeles Rams played in St. Louis, Missouri, as the St. Louis Rams from the 1995 through the 2015 seasons before relocating back to Los Angeles where the team had played from the 1946 season to the 1994 season. The Rams franchise relocated from Los Angeles to St. Louis in 1995, which had been without a National Football League (NFL) team since the Cardinals moved to Phoenix, Arizona in 1988. The Rams' first home game in St. Louis was at Busch Memorial Stadium against the
Document [4](Title: Los Angeles Rams): in 1980. After the 1994 NFL season, the Rams left California and moved east to St. Louis, Missouri. Five seasons after relocating, the team won Super Bowl XXXIV in a 23–16 victory over the Tennessee Titans. They appeared again in Super Bowl XXXVI, where they lost 20–17 to the New England Patriots. The Rams continued to play in Edward Jones Dome in St. Louis until the end of the 2015 NFL season, when the team filed notice with the NFL of its intent to pursue a relocation back to Los Angeles. The move was approved by a 30–2 margin at
Document [5](Title: 1994 Los Angeles Rams season): 1994 Los Angeles Rams season The 1994 Los Angeles Rams season was the franchise's 57th year with the National Football League and the 49th and last season in the Greater Los Angeles Area until their 2016 relocation back to Los Angeles. After nearly 50 years in the Greater Los Angeles Area, owner Georgia Frontiere announced that the team would relocate to St. Louis, Missouri on January 15, 1995. While the owners initially rejected the move, permission was eventually granted therefore bringing an end to Southern California's first major professional sports franchise until 2016. The threat of relocation dominated talk about

Output
...Their first home game in St. Louis was at Busch Memorial Stadium against the Chicago Bears on October 22,1996 [1]....

Table 15: Sample containing fabrication error. In this sample, **Chicago Bears** does not appear in the reference documents.

Question
Who performed at the champions league final 2018?
Documents
Document [1](Title: 2016 UEFA Champions League Final): worldwide via UEFA.com from 1 to 14 March 2016 in four price categories: €440, €320, €160 and €70. The remaining tickets were allocated to the local organising committee, UEFA and national associations, commercial partners and broadcasters, and to serve the corporate hospitality programme. American singer Alicia Keys performed in the opening ceremony prior to the match, the first time it has featured a live music performance. Italian tenor Andrea Bocelli performed the UEFA Champions League Anthem. The 2016 UEFA Women's Champions League Final was held two days prior, on 26 May 2016, at the Mapei Stadium – Città del Tricolore
Document [2](Title: 2018 UEFA Champions League Final): Lipa performed at the opening ceremony preceding the final. Jamaican rapper Sean Paul joined her as a special guest to perform their collaborative song, N̈o Lie. The 2018 UEFA Women's Champions League Final was held two days earlier, on 24 May 2018, at the Valeriy Lobanovskyi Dynamo Stadium between Wolfsburg and Lyon, Lyon emerging victorious 4–1. This was also the last time that the host city for the men's Champions League final was also automatically assigned the Women's Champions League final. The annual UEFA Champions Festival was held between 24–27 May 2018 at the Kiev city centre. In late May,
Document [3](Title: UEFA Champions League Anthem): the two teams are lined up, as well as at the beginning and end of television broadcasts of the matches. Special vocal versions have been performed live at the Champions League Final with lyrics in other languages, changing over to the host country's language for the chorus. These versions were performed by Andrea Bocelli (Italian) (Rome 2009, Milan 2016 and Cardiff 2017), Juan Diego Flores (Spanish) (Madrid 2010), All Angels (Wembley 2011), Jonas Kaufmann and David Garrett (Munich 2012), Mariza (Lisbon 2014, unlike the previous final performers, Mariza sang the main lyric of the anthem), and Nina Maria Fischer and
Document [4](Title: 2018 UEFA Champions League Final): 2018 UEFA Champions League Final The 2018 UEFA Champions League Final was the final match of the 2017–18 UEFA Champions League, the 63rd season of Europe's premier club football tournament organised by UEFA, and the 26th season since it was renamed from the European Cup to the UEFA Champions League. It was played at the NSC Olimpiyskiy Stadium in Kiev, Ukraine on 26 May 2018, between Spanish side and defending champions Real Madrid, who had won the competition in each of the last two seasons, and English side Liverpool. Real Madrid won the final 3–1 for their third consecutive and
Document [5](Title: 2017 UEFA Champions League Final): 2017 UEFA Champions League Final The 2017 UEFA Champions League Final was the final match of the 2016–17 UEFA Champions League, the 62nd season of Europe's premier club football tournament organised by UEFA, and the 25th season since it was renamed from the European Champion Clubs' Cup to the UEFA Champions League. It was played at the Millennium Stadium in Cardiff, Wales on 3 June 2017, between Italian side Juventus and Spanish side and title holders Real Madrid, in a repeat of the 1998 final. Real Madrid won the match 4–1 to secure their 12th title in this competition. With

Output
...Real Madrid won the final 3-1 for their third consecutive and overall 12th title in this competition[4]...

Table 16: Sample containing synthesis error. In this sample, document 4 and document 5 are mistakenly intermingled.

Question
When was the 13th amendment ratified by the states?
Documents
Document [1](Title: Thirteenth Amendment to the United States Constitution): ratification did not imply federal power to legislate on the status of former slaves. During the first week of December, North Carolina and Georgia gave the amendment the final votes needed for it to become part of the Constitution. The Thirteenth Amendment became part of the Constitution on December 6, 1865, based on the following ratifications: Having been ratified by the legislatures of three-fourths of the several states (27 of the 36 states, including those that had been ïn rebellion¨), Secretary of State Seward, on December 18, 1865, certified that the Thirteenth Amendment had become valid, to all intents and
Document [2](Title: Thirteenth Amendment to the United States Constitution): Thirteenth Amendment to the United States Constitution The Thirteenth Amendment (Amendment XIII) to the United States Constitution abolished slavery and involuntary servitude, except as punishment for a crime. In Congress, it was passed by the Senate on April 8, 1864, and by the House on January 31, 1865. The amendment was ratified by the required number of states on December 6, 1865. On December 18, 1865, Secretary of State William H. Seward proclaimed its adoption. It was the first of the three Reconstruction Amendments adopted following the American Civil War. Since the American Revolution, states had divided into states that
Document [3](Title: Emancipation Proclamation): Winning re-election, Lincoln pressed the lame duck 38th Congress to pass the proposed amendment immediately rather than wait for the incoming 39th Congress to convene. In January 1865, Congress sent to the state legislatures for ratification what became the Thirteenth Amendment, banning slavery in all U.S. states and territories. The amendment was ratified by the legislatures of enough states by December 6, 1865, and proclaimed 12 days later. There were about 40,000 slaves in Kentucky and 1,000 in Delaware who were liberated then. As the years went on and American life continued to be deeply unfair towards blacks, cynicism towards
Document [4](Title: Thirteenth Amendment to the United States Constitution): Enforcement, and Contemporary Implications Thirteenth Amendment to the United States Constitution The Thirteenth Amendment (Amendment XIII) to the United States Constitution abolished slavery and involuntary servitude, except as punishment for a crime. In Congress, it was passed by the Senate on April 8, 1864, and by the House on January 31, 1865. The amendment was ratified by the required number of states on December 6, 1865. On December 18, 1865, Secretary of State William H. Seward proclaimed its adoption. It was the first of the three Reconstruction Amendments adopted following the American Civil War. Since the American Revolution, states had
Document [5](Title: Thirteenth Amendment to the United States Constitution): joint resolution was sent to Lincoln for his signature. Under the usual signatures of the Speaker of the House and the President of the Senate, President Lincoln wrote the word Äpprovedänd added his signature to the joint resolution on February 1, 1865. On February 7, Congress passed a resolution affirming that the Presidential signature was unnecessary. The Thirteenth Amendment is the only ratified amendment signed by a President, although James Buchanan had signed the Corwin Amendment that the 36th Congress had adopted and sent to the states in March 1861. When the Thirteenth Amendment was submitted to the states

Output
...The amendment was ratified by the required number of states on December 6,1865, and President Lincoln signed the joint resolution on February 1,1865, after it was passed by Congress on February 7,1865 [5]...

Table 17: Sample containing omission error. In this sample, document 4 is not attributed.

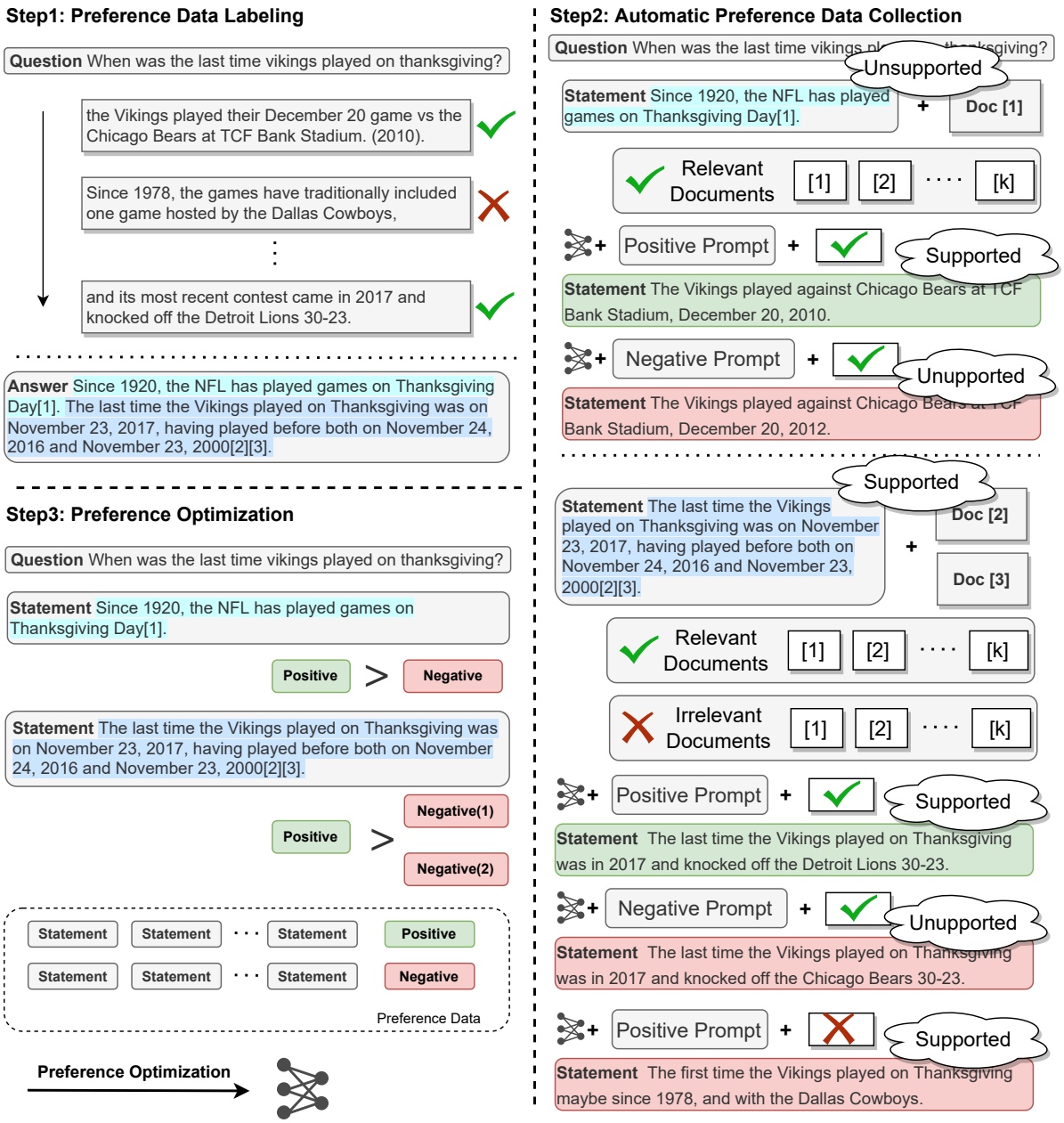

Figure 2: The overall framework of the APO.