SELF-RAG: LEARNING TO RETRIEVE, GENERATE, AND CRITIQUE THROUGH SELF-REFLECTION

Akari Asai[†], **Zeqiu Wu**[†], **Yizhong Wang**[†§], **Avirup Sil**[‡], **Hannaneh Hajishirzi**[†§]
[†]University of Washington [§]Allen Institute for AI [‡]IBM Research AI
{akari,zeqiuwu1,yizhongw,hannaneh}@cs.washington.edu, avi@us.ibm.com

ABSTRACT

Despite their remarkable capabilities, large language models (LLMs) often produce responses containing factual inaccuracies due to their sole reliance on the parametric knowledge they encapsulate. Retrieval-Augmented Generation (RAG), an ad hoc approach that augments LMs with retrieval of relevant knowledge, decreases such issues. However, indiscriminately retrieving and incorporating a fixed number of retrieved passages, regardless of whether retrieval is necessary, or passages are relevant, diminishes LM versatility or can lead to unhelpful response generation. We introduce a new framework called **Self-Reflective Retrieval-Augmented Generation (SELF-RAG)** that enhances an LM's quality and factuality through retrieval and self-reflection. Our framework trains a single arbitrary LM that adaptively retrieves passages on-demand, and generates and reflects on retrieved passages and its own generations using special tokens, called *reflection* tokens. Generating reflection tokens makes the LM controllable during the inference phase, enabling it to tailor its behavior to diverse task requirements. Experiments show that SELF-RAG (7B and 13B parameters) significantly outperforms state-of-the-art LLMs and retrieval-augmented models on a diverse set of tasks. Specifically, SELF-RAG outperforms ChatGPT and retrieval-augmented Llama2-chat on Open-domain QA, reasoning and fact verification tasks, and it shows significant gains in improving factuality and citation accuracy for long-form generations relative to these models.[1]

1 INTRODUCTION

State-of-the-art LLMs continue to struggle with factual errors (Mallen et al., 2023; Min et al., 2023) despite their increased model and data scale (Ouyang et al., 2022). Retrieval-Augmented Generation (RAG) methods (Figure 1 left; Lewis et al. 2020; Guu et al. 2020) augment the input of LLMs with relevant retrieved passages, reducing factual errors in knowledge-intensive tasks (Ram et al., 2023; Asai et al., 2023a). However, these methods may hinder the versatility of LLMs or introduce unnecessary or off-topic passages that lead to low-quality generations (Shi et al., 2023) since they retrieve passages indiscriminately regardless of whether the factual grounding is helpful. Moreover, the output is not guaranteed to be consistent with retrieved relevant passages (Gao et al., 2023) since the models are not explicitly trained to leverage and follow facts from provided passages. This work introduces **Self-Reflective Retrieval-augmented Generation (SELF-RAG)** to improve an LLM's generation quality, including its factual accuracy without hurting its versatility, via on-demand retrieval and self-reflection. We train an arbitrary LM in an end-to-end manner to learn to reflect on its own generation process given a task input by generating both task output and intermittent special tokens (i.e., *reflection tokens*). Reflection tokens are categorized into *retrieval* and *critique* tokens to indicate the need for retrieval and its generation quality respectively (Figure 1 right). In particular, given an input prompt and preceding generations, SELF-RAG first determines if augmenting the continued generation with retrieved passages would be helpful. If so, it outputs a **retrieval** token that calls a retriever model on demand (Step 1). Subsequently, SELF-RAG concurrently processes multiple retrieved passages, evaluating their relevance and then **generating** corresponding task outputs (Step 2). It then generates critique tokens to **criticize** its own output and choose best one (Step 3) in terms of factuality and overall quality. This process differs from conventional RAG (Figure 1 left), which

[1]Our code and trained models are available at https://selfrag.github.io/.

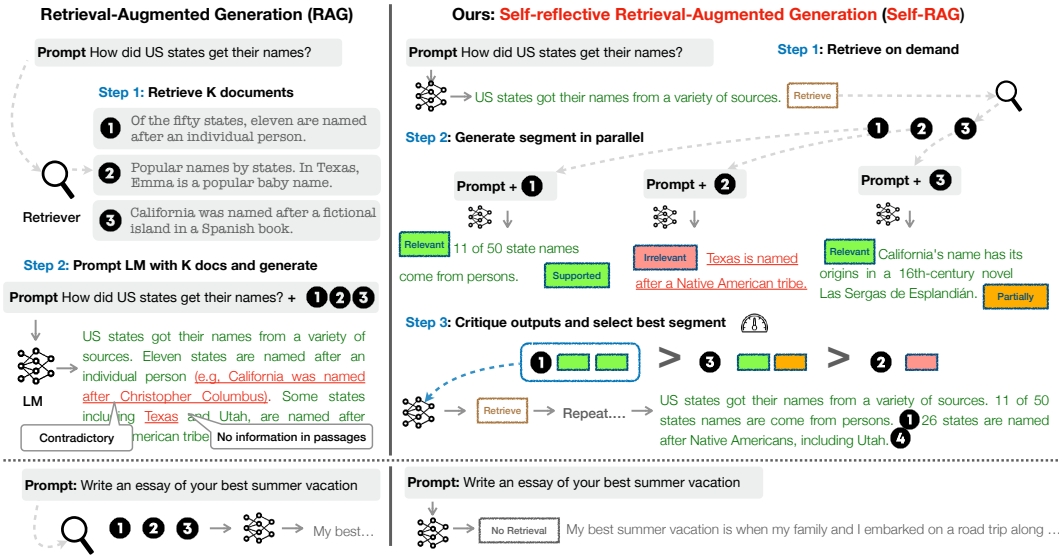


Figure 1: Overview of SELF-RAG. SELF-RAG learns to retrieve, critique, and generate text passages to enhance overall generation quality, factuality, and verifiability.

consistently retrieves a fixed number of documents for generation regardless of the retrieval necessity (e.g., the bottom figure example does not require factual knowledge) and never second visits the generation quality. Moreover, SELF-RAG provides citations for each segment with its self-assessment of whether the output is supported by the passage, leading to easier fact verification.

SELF-RAG trains an arbitrary LM to generate text with reflection tokens by unifying them as the next token prediction from the expanded model vocabulary. We train our generator LM on a diverse collection of text interleaved with reflection tokens and retrieved passages. Reflection tokens, inspired by reward models used in reinforcement learning (Ziegler et al., 2019; Ouyang et al., 2022), are inserted offline into the original corpus by a trained *critic* model. This eliminates the need to host a critic model during training, reducing overhead. The critic model, in part, is supervised on a dataset of input, output, and corresponding reflection tokens collected by prompting a propriety LM (i.e., GPT-4; OpenAI 2023). While we draw inspiration from studies that use control tokens to start and guide text generation (Lu et al., 2022; Keskar et al., 2019), our trained LM uses critique tokens to assess its own predictions after each generated segment as an integral part of the generation output.

SELF-RAG further enables a customizable decoding algorithm to satisfy hard or soft constraints, which are defined by reflection token predictions. In particular, our inference-time algorithm enables us to (1) flexibly adjust retrieval frequency for different downstream applications and (2) customize models' behaviors to user preferences by leveraging reflection tokens through segment-level beam search using the weighted linear sum of the reflection token probabilities as segment score.

Empirical results on six tasks, including reasoning and long-form generation, demonstrate that SELF-RAG significantly outperforms pre-trained and instruction-tuned LLMs that have more parameters and widely adopted RAG approaches with higher citation accuracy. In particular, SELF-RAG outperforms retrieval-augmented ChatGPT on four tasks, Llama2-chat (Touvron et al., 2023) and Alpaca (Dubois et al., 2023) on all tasks. Our analysis demonstrates the effectiveness of training and inference with reflection tokens for overall performance improvements as well as test-time model customizations (e.g., balancing the trade-off between citation previsions and completeness).

2 RELATED WORK

Retrieval-Augmented Generation. Retrieval-Augmented Generation (RAG) augments the input space of LMs with retrieved text passages (Guu et al., 2020; Lewis et al., 2020), leading to large improvements in knowledge-intensive tasks after fine-tuning or used with off-the-shelf LMs (Ram et al., 2023). A more recent work (Luo et al., 2023) instruction-tunes an LM with a fixed number

of retrieved passages prepended to input, or pre-train a retriever and LM jointly, followed by few-shot fine-tuning on task datasets (Izacard et al., 2022b). While prior work often retrieves only once at the beginning, Jiang et al. (2023) propose to adaptively retrieve passages for generation on top of a proprietary LLM or Schick et al. (2023) train an LM to generate API calls for named entities. Yet, the improved task performance of such approaches often comes at the expense of runtime efficiency (Mallen et al., 2023), robustness to irrelevant context (Shi et al., 2023), and lack of attributions (Liu et al., 2023a; Gao et al., 2023). We introduce a method to train an arbitrary LM to learn to use retrieval *on-demand* for diverse instruction-following queries and introduce controlled generation guided by reflections tokens to further improve generation quality and attributions.

Concurrent RAG work. A few concurrent works[2] on RAG propose new training or prompting strategies to improve widely-adopted RAG approaches. Lin et al. (2023) fine-tune both the retriever and LM on instruction-tuning datasets in two steps. While we also train our model on diverse instruction-following datasets, SELF-RAG enables retrieval on demand and selection of the best possible model output via fine-grained self-reflection, making it widely applicable and more robust and controllable. Yoran et al. (2023) use a natural language inference model and Xu et al. (2023) use a summarization model to filter out or compress retrieved passages before using them to prompt the LM to generate the output. SELF-RAG processes passages in parallel and filters out irrelevant ones through self-reflection, without relying on external models at inference. Moreover, our self-reflection mechanism also evaluates other aspects of the model output quality including factuality. LATS (Zhou et al., 2023) prompt off-the-shelf LMs to search for relevant information for question answering tasks and to generate with tree search, guided by LM-generated value scores. While their value function simply indicates an overall score of each generation, SELF-RAG trains to an arbitrary LM to learn to generate fine-grained self-reflection and customizable inference.

Training and generating with critics. Training LLMs with reinforcement learning (e.g., Proximal Policy Optimization or PPO; Schulman et al. 2017) from human feedback (RLHF) has proven effective in aligning LLMs with human preferences (Ouyang et al., 2022). Wu et al. (2023) introduce fine-grained RLHF with multiple reward models. Though our work also studies fine-grained critique on retrieval and generation, we train our target LM on task examples augmented with reflection tokens from a critic model offline, with a far lower training cost compared to RLHF. In addition, reflection tokens in SELF-RAG enable controllable generation at inference, while RLHF focuses on human preference alignment during training. Other works use general control tokens to guide LM generation (Lu et al., 2022; Korbak et al., 2023), while SELF-RAG uses reflection tokens to decide the need for retrieval and to self-evaluate generation quality. Xie et al. (2023) propose a self-evaluation-guided decoding framework, but they focus only on reasoning tasks with one evaluation dimension (reasoning path consistency) and without retrieval. Recent work on LLM refinement (Dhuliawala et al., 2023; Madaan et al., 2023; Paul et al., 2023) prompts a model to generate task output, natural language feedback and refined task output iteratively, but at the cost of inference efficiency.

3 SELF-RAG: LEARNING TO RETRIEVE, GENERATE AND CRITIQUE

We introduce Self-Reflective Retrieval-Augmented Generation (SELF-RAG), shown in Figure 1. SELF-RAG is a framework that enhances the quality and factuality of an LLM through retrieval and self-reflection, without sacrificing LLM's original creativity and versatility. Our end-to-end training lets an LM $\mathcal{M}$ **generate** text informed by **retrieved** passages, if needed, and **criticize** the output by learning to generate special tokens. These *reflection tokens* (Table 1) signal the need for retrieval or confirm the output's relevance, support, or completeness. In contrast, common RAG approaches retrieve passages indiscriminately, without ensuring complete support from cited sources.

3.1 PROBLEM FORMALIZATION AND OVERVIEW

Formally, given input x, we train $\mathcal{M}$ to sequentially generate textual outputs y consisting of multiple segments $y = [y_1, \ldots, y_T]$, where y_t indicates a sequence of tokens for the t-th segment.[3] Generated tokens in y_t include text from the original vocabulary as well as the reflection tokens (Table 1).

[2]All work is arXived within a week of this preprint.

[3]In this paper, we treat one sentence as a segment in our experiments, but our framework is applicable to any segment unit (i.e., sub-sentence).

Type	Input	Output	Definitions
Retrieve	x / x, y	{yes, no, continue}	Decides when to retrieve with $\mathcal{R}$
IsRel	x, d	{**relevant**, irrelevant}	d provides useful information to solve x.
IsSup	x, d, y	{**fully supported**, partially supported, no support}	All of the verification-worthy statement in y is supported by d.
IsUse	x, y	{**5**, 4, 3, 2, 1}	y is a useful response to x.

Table 1: Four types of reflection tokens used in SELF-RAG. Each type uses several tokens to represent its output values. The bottom three rows are three types of Critique tokens, and **the bold text** indicates the most desirable critique tokens. x, y, d indicate input, output, and a relevant passage, respectively.

Algorithm 1 SELF-RAG Inference

Require: Generator LM $\mathcal{M}$, Retriever $\mathcal{R}$, Large-scale passage collections $\{d_1, \ldots, d_N\}$
Input: input prompt x and preceding generation $y_{<t}$, **Output:** next output segment y_t
$\mathcal{M}$ predicts Retrieve given $(x, y_{<t})$
if Retrieve == Yes **then**
  Retrieve relevant text passages $\mathbf{D}$ using $\mathcal{R}$ given (x, y_{t-1}) ▷ Retrieve
  $\mathcal{M}$ predicts IsRel given x, d and y_t given $x, d, y_{<t}$ for each $d \in \mathbf{D}$ ▷ Generate
  $\mathcal{M}$ predicts IsSup and IsUse given x, y_t, d for each $d \in \mathbf{D}$ ▷ Critique
  Rank y_t based on IsRel , IsSup , IsUse ▷ Detailed in Section 3.3
else if Retrieve == No **then**
  $\mathcal{M}_{gen}$ predicts y_t given x ▷ Generate
  $\mathcal{M}_{gen}$ predicts IsUse given x, y_t ▷ Critique

Inference overview. Figure 1 and Algorithm 1 present an overview of SELF-RAG at inference. For every x and preceding generation $y_{<t}$, the model decodes a retrieval token to evaluate the utility of retrieval. If retrieval is not required, the model predicts the next output segment, as it does in a standard LM. If retrieval is needed, the model generates: a critique token to evaluate the retrieved passage's relevance, the next response segment, and a critique token to evaluate if the information in the response segment is supported by the passage. Finally, a new critique token evaluates the overall utility of the response.[4] To generate each segment, SELF-RAG processes multiple passages in parallel and uses its own generated reflection tokens to enforce soft constraints (Section 3.3) or hard control (Algorithm 1) over the generated task output. For instance, in Figure 1 (right), the retrieved passages d_1 is selected at the first time step since d_2 does not provide direct evidence (IsRel is Irrelevant) and d_3 output is only partially supported while d_1 are fully supported.

Training overview. SELF-RAG enables an arbitrary LM to generate text with reflection tokens by unifying them as next token predictions from the expanded model vocabulary (i.e., the original vocabulary plus reflection tokens). Specifically, we train the generator model $\mathcal{M}$ on a curated corpus with interleaving passages retrieved by a *retriever* $\mathcal{R}$ and reflection tokens predicted by a *critic* model $\mathcal{C}$ (summarized in Appendix Algorithm 2). We train $\mathcal{C}$ to generate reflection tokens for evaluating retrieved passages and the quality of a given task output (Section 3.2.1). Using the critic model, we update the training corpus by inserting reflection tokens into task outputs offline. Subsequently, we train the final generator model ($\mathcal{M}$) using the conventional LM objective (Section 3.2.2) to enable $\mathcal{M}$ to generate reflection tokens by itself without relying on the critic at inference time.

3.2 SELF-RAG TRAINING

Here, we describe the supervised data collection and training of two models, the critic $\mathcal{C}$ (Section 3.2.1) and the generator $\mathcal{M}$ (Section 3.2.2).

3.2.1 TRAINING THE CRITIC MODEL

Data collection for critic model. Manual annotation of reflection tokens for each segment is expensive (Wu et al., 2023). A state-of-the-art LLM like GPT-4 (OpenAI, 2023) can be effectively

[4]We follow Liu et al. (2023a) in using a "perceived" utility value that is independent of retrieved passages.

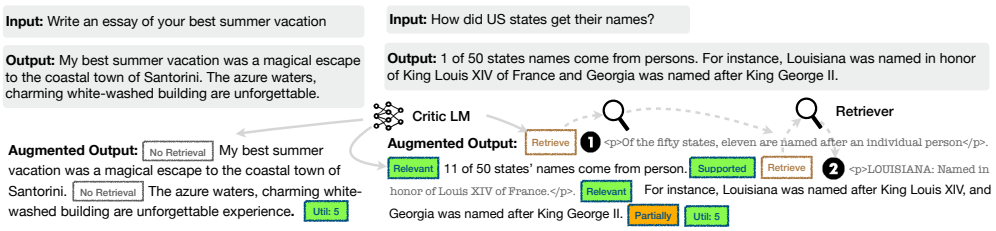


Figure 2: SELF-RAG training examples. The left example does not require retrieval while the right one requires retrieval; thus, passages are inserted. More examples are in Appendix Table 4.

used to generate such feedback (Liu et al., 2023b). However, depending on such proprietary LMs can raise API costs and diminish reproducibility (Chen et al., 2023). We create supervised data by prompting GPT-4 to generate reflection tokens and then distill their knowledge into an in-house $\mathcal{C}$. For each group of reflection tokens, we randomly sample instances from the original training data: $\{X^{sample}, Y^{sample}\} \sim \{X, Y\}$. As different reflection token groups have their own definitions and input, as shown in Table 1, we use different instruction prompts for them. Here, we use Retrieve as an example. We prompt GPT-4 with a type-specific instruction ("Given an instruction, make a judgment on whether finding some external documents from the web helps to generate a better response.") followed by few-shot demonstrations I the original task input x and output y to predict an appropriate reflection token as text: $p(r|I, x, y)$. Manual assessment reveals that GPT-4 reflection token predictions show high agreement with human evaluations. We collect 4k-20k supervised training data for each type and combine them to form training data for $\mathcal{C}$. Appendix Section D shows the full list of instructions, and A.1 contains more details and our analysis.

Critic learning. After we collect training data $\mathcal{D}_{critic}$, we initialize $\mathcal{C}$ with a pre-trained LM and train it on $\mathcal{D}_{critic}$ using a standard conditional language modeling objective, maximizing likelihood:

$$\max_{\mathcal{C}} \mathbb{E}_{((x,y),r)\sim\mathcal{D}_{critic}} \log p_{\mathcal{C}}(r|x, y), \ r \text{ for reflection tokens.} \tag{1}$$

Though the initial model can be any pre-trained LM, we use the same one as the generator LM (i.e., Llama 2-7B; Touvron et al. 2023) for $\mathcal{C}$ initialization. The critic achieves a higher than 90% agreement with GPT-4-based predictions on most reflection token categories (Appendix Table 5).

3.2.2 TRAINING THE GENERATOR MODEL

Data collection for generator. Given an input-output pair (x, y), we augment the original output y using the retrieval and critic models to create supervised data that precisely mimics the SELF-RAG inference-time process (Section 3.1). For each segment $y_t \in y$, we run $\mathcal{C}$ to assess whether additional passages could help to enhance generation. If retrieval is required, the retrieval special token Retrieve =Yes is added, and $\mathcal{R}$ retrieves the top K passages, **D**. For each passage, $\mathcal{C}$ further evaluates whether the passage is relevant and predicts IsRel. If a passage is relevant, $\mathcal{C}$ further evaluates whether the passage supports the model generation and predicts IsSup. Critique tokens IsRel and IsSup are appended after the retrieved passage or generations. At the end of the output, y (or y_T), $\mathcal{C}$ predicts the overall utility token IsUse, and an augmented output with reflection tokens and the original input pair is added to $\mathcal{D}_{gen}$. See the example training data in Figure 2.

Generator learning. We train the generator model $\mathcal{M}$ by training on the curated corpus augmented with reflection tokens $\mathcal{D}_{gen}$ using the standard next token objective:

$$\max_{\mathcal{M}} \mathbb{E}_{(x,y,r)\sim\mathcal{D}_{gen}} \log p_{\mathcal{M}}(y, r|x). \tag{2}$$

Unlike $\mathcal{C}$ training (Eq. 1), $\mathcal{M}$ learns to predict the target output as well as the reflection tokens. During training, we mask out the retrieved text chunks (surrounded by and in Figure 2) for loss calculation and expand the original vocabulary $\mathcal{V}$ with a set of reflection tokens {Critique, Retrieve}.

Connections to prior work on learning with critique. Recent work incorporates additional critique (feedback) during training, e.g., RLHF (Ouyang et al. 2022) via PPO. While PPO relies on

separate reward models during training, we compute critique offline and directly insert them into the training corpus, where the generator LM is trained with a standard LM objective. This significantly reduces training costs compared to PPO. Our work also relates to prior work that incorporates special tokens to control generation (Keskar et al., 2019; Lu et al., 2022; Korbak et al., 2023). Our SELF-RAG learns to generate special tokens *to evaluate its own prediction* after each generated segment, enabling the use of a soft re-ranking mechanism or hard constraints at inference (discussed next).

3.3 SELF-RAG INFERENCE

Generating reflection tokens to self-evaluate its own output makes SELF-RAG controllable during the inference phase, enabling it to tailor its behavior to diverse task requirements. For tasks demanding factual accuracy (Min et al., 2023), we aim for the model to retrieve passages more frequently to ensure that the output aligns closely with the available evidence. Conversely, in more open-ended tasks, like composing a personal experience essay, the emphasis shifts towards retrieving less and prioritizing the overall creativity or utility score. In this section, we describe approaches to enforce control to meet these distinct objectives during the inference process.

Adaptive retrieval with threshold. SELF-RAG dynamically decides when to retrieve text passages by predicting [Retrieve]. Alternatively, our framework allows a threshold to be set. Specifically, if the probability of generating the [Retrieve]=`Yes` token normalized over all output tokens in [Retrieve] surpasses a designated threshold, we trigger retrieval (details in Appendix Section A.3).

Tree-decoding with critique tokens. At each segment step t, when retrieval is required, based either on hard or soft conditions, $\mathcal{R}$ retrieves K passages, and the generator $\mathcal{M}$ processes each passage in parallel and outputs K different continuation candidates. We conduct a segment-level beam search (with the beam size=B) to obtain the top-B segment continuations at each timestamp t, and return the best sequence at the end of generation. The score of each segment y_t with respect to passage d is updated with a critic score $\mathcal{S}$ that is the linear weighted sum of the normalized probability of each [Critique] token type. For each critique token group G (e.g., [IsRel]), we denote its score at timestamp t as s_t^G, and we compute a segment score as follows:

$$f(y_t, d, [\text{Critique}]) = p(y_t|x, d, y_{<t})) + \mathcal{S}([\text{Critique}]), \text{where} \tag{3}$$

$$\mathcal{S}([\text{Critique}]) = \sum_{G \in \mathcal{G}} w^G s_t^G \text{ for } \mathcal{G} = \{[\text{IsRel}], [\text{IsSup}], [\text{IsUse}]\}, \tag{4}$$

where $s_t^G = \frac{p_t(\hat{r})}{\sum_{i=1}^{N^G} p_t(r_i)}$ stands for the generation probability of the most desirable reflection token $\hat{r}$ (e.g., [IsRel] =`Relevant`) for the critique token type G with N^G distinct tokens (that represent different possible values for G). The weights w^G in Eq. 4 are hyperparameters that can be adjusted at inference time to enable customized behaviors at test time. For instance, to ensure that result y is mostly supported by evidence, we can set a weight term for the [IsSup] score higher, while relatively lowering weights for other aspects. Alternatively, we could further enforce hard constraints during decoding using [Critique]. Instead of using a soft reward function in Eq. 4, we could explicitly filter out a segment continuation when the model generates an undesirable [Critique] token (e.g., [IsSup] =`No support`) . Balancing the trade-off between multiple preferences has been studied in RLHF (Touvron et al., 2023; Wu et al., 2023), which often requires training to change models' behaviors. SELF-RAG tailors an LM with no additional training.

4 EXPERIMENTS

4.1 TASKS AND DATASETS

We conduct evaluations of our SELF-RAG and diverse baselines on a range of downstream tasks, holistically evaluating outputs with metrics designed to assess overall correctness, factuality, and fluency. Throughout these experiments, we conduct zero-shot evaluations, where we provide instructions describing tasks without few-shot demonstrations (Wei et al., 2022; Sanh et al., 2022). Details of our experiments' settings, including test-time instructions, are available in the Appendix Section B.1.

Closed-set tasks include two datasets, i.e., a fact *verification dataset* about public health (**PubHealth**; Zhang et al. 2023) and a *multiple-choice reasoning dataset* created from scientific exams (**ARC-**

Challenge; Clark et al. 2018). We use accuracy as an evaluation metric and report on the test set. We aggregate the answer probabilities of target classes for both of these datasets (Appendix Section B.2).

Short-form generations tasks include two open-domain question answering (QA) datasets, PopQA (Mallen et al., 2023) and TriviaQA-unfiltered (Joshi et al., 2017), where systems need to answer arbitrary questions about factual knowledge. For PopQA, we use the long-tail subset, consisting of 1,399 rare entity queries whose monthly Wikipedia page views are less than 100. As the TriviaQA-unfiltered (open) test set is not publicly available, we follow prior work's validation and test split (Min et al., 2019; Guu et al., 2020), using 11,313 test queries for evaluation. We evaluate performance based on whether gold answers are included in the model generations instead of strictly requiring exact matching, following Mallen et al. (2023); Schick et al. (2023).

Long-form generation tasks include a biography generation task (Min et al., 2023) and a long-form QA task **ALCE-ASQA** Gao et al. (2023); Stelmakh et al. (2022). We use FactScore (Min et al., 2023) to evaluate biographies, and we use official metrics of correctness (str-em), fluency based on MAUVE (Pillutla et al., 2021), and citation precision and recall (Gao et al., 2023) for ASQA. [5]

4.2 BASELINES

Baselines without retrievals. We evaluate strong publicly available pre-trained LLMs, $\text{Llama2}_{7\text{B},13\text{B}}$ (Touvron et al., 2023), instruction-tuned models, $\text{Alpaca}_{7\text{B},13\text{B}}$ (Dubois et al., 2023) (our replication based on Llama2); and models trained and reinforced using private data, ChatGPT (Ouyang et al., 2022) and $\text{Llama2-chat}_{13\text{B}}$. For instruction-tuned LMs, we use the official system prompt or instruction format used during training if publicly available. We also compare our method to concurrent work, $\text{CoVE}_{65\text{B}}$ (Dhuliawala et al., 2023), which introduces iterative prompt engineering to improve the factuality of LLM generations.

Baselines with retrievals. We evaluate models augmented with retrieval at test time or during training. The first category includes standard RAG baselines, where an LM (Llama2, Alpaca) generates output given the query prepended with the top retrieved documents using the same retriever as in our system. It also includes Llama2-FT, where Llama2 is fine-tuned on all training data we use without the reflection tokens or retrieved passages. We also report the result of retrieval-augmented baselines with LMs trained with private data: Ret-ChatGPT and Ret-Llama2-chat, which deploy the same augmentation technique above, as well as perplexity.ai, an InstructGPT-based production search system. The second category includes concurrent methods that are trained with retrieved text passages, i.e., SAIL (Luo et al., 2023) to instruction-tune an LM on the Alpaca instruction-tuning data with top retrieved documents inserted before instructions, and Toolformer (Schick et al., 2023) to pre-train an LM with API calls (e.g., Wikipedia APIs).[6]

4.3 EXPERIMENTAL SETTINGS

Training data and settings. Our training data consists of diverse instruction-following input-output pairs. In particular, we sample instances from Open-Instruct processed data (Wang et al., 2023) and knowledge-intensive datasets (Petroni et al., 2021; Stelmakh et al., 2022; Mihaylov et al., 2018). In total, we use 150k instruction-output pairs. We use Llama2 7B and 13B (Touvron et al., 2023) as our generator base LM, and we use Llama2 7B as our base critic LM. For the retriever model $\mathcal{R}$, we use off-the-shelf Contriever-MS MARCO (Izacard et al., 2022a) by default and retrieve up to ten documents for each input. More training details are in the Appendix Section B.1.

Inference settings. As a default configuration, we assign the weight terms IsRel, IsSup, IsUse values of 1.0, 1.0 and 0.5, respectively. To encourage frequent retrieval, we set the retrieval threshold to 0.2 for most tasks and to 0 for ALCE (Gao et al., 2023) due to citation requirements. We speed up inference using vllm (Kwon et al., 2023). At each segment level, we adopt a beam width of 2. For a token-level generation, we use greedy decoding. By default, we use the top five documents from Contriever-MS MARCO (Izacard et al., 2022a); for biographies and open-domain QA, we use additional top five documents retrieved by a web search engine, following Luo et al. (2023); for ASQA, we use the author-provided top 5 documents by GTR-XXL (Ni et al., 2022) across all baselines for a fair comparison.

[5] https://github.com/princeton-nlp/ALCE

[6] We report numbers using the results reported in the paper as the implementations are not available.

Table 2: Overall experiment results on six tasks. **Bold** numbers indicate the best performance among non-proprietary models, and **gray-colored** bold text indicates the best proprietary model when they outperforms all non-proprietary models. * indicates concurrent or recent results reported by concurrent work. – indicates numbers that are not reported by the original papers or are not applicable. Models are sorted based on scale. FS, em, rg, mau, prec, rec denote FactScore (factuality); str-em, rouge (correctness); MAUVE (fluency); citation precision and recall, respectively.

	Short-form		Closed-set		Long-form generations (with citations)					
	PopQA	TQA	Pub	ARC	Bio	ASQA				
LM	(acc)	(acc)	(acc)	(acc)	(FS)	(em)	(rg)	(mau)	(pre)	(rec)
LMs with proprietary data										
Llama2-c$_{13B}$	20.0	59.3	49.4	38.4	55.9	22.4	29.6	28.6	–	–
Ret-Llama2-c$_{13B}$	51.8	59.8	52.1	37.9	79.9	32.8	34.8	43.8	19.8	36.1
ChatGPT	29.3	**74.3**	70.1	**75.3**	71.8	35.3	36.2	68.8	–	–
Ret-ChatGPT	50.8	65.7	54.7	**75.3**	–	**40.7**	**39.9**	**79.7**	65.1	**76.6**
Perplexity.ai	–	–	–	–	71.2	–	–	–	–	–
Baselines without retrieval										
Llama2$_{7B}$	14.7	30.5	34.2	21.8	44.5	7.9	15.3	19.0	–	–
Alpaca$_{7B}$	23.6	54.5	49.8	45.0	45.8	18.8	29.4	61.7	–	–
Llama2$_{13B}$	14.7	38.5	29.4	29.4	53.4	7.2	12.4	16.0	–	–
Alpaca$_{13B}$	24.4	61.3	55.5	54.9	50.2	22.9	32.0	70.6	–	–
CoVE$_{65B}$ *	–	–	–	–	71.2	–	–	–	–	–
Baselines with retrieval										
Toolformer*$_{6B}$	–	48.8	–	–	–	–	–	–	–	–
Llama2$_{7B}$	38.2	42.5	30.0	48.0	78.0	15.2	22.1	32.0	2.9	4.0
Alpaca$_{7B}$	46.7	64.1	40.2	48.0	76.6	30.9	33.3	57.9	5.5	7.2
Llama2-FT$_{7B}$	48.7	57.3	64.3	65.8	78.2	31.0	35.8	51.2	5.0	7.5
SAIL*$_{7B}$	–	–	69.2	48.4	–	–	–	–	–	–
Llama2$_{13B}$	45.7	47.0	30.2	26.0	77.5	16.3	20.5	24.7	2.3	3.6
Alpaca$_{13B}$	46.1	66.9	51.1	57.6	77.7	**34.8**	36.7	56.6	2.0	3.8
Our SELF-RAG $_{7B}$	54.9	66.4	72.4	67.3	**81.2**	30.0	35.7	**74.3**	66.9	67.8
Our SELF-RAG $_{13B}$	**55.8**	**69.3**	**74.5**	**73.1**	80.2	31.7	**37.0**	71.6	**70.3**	**71.3**

5 RESULTS AND ANALYSIS

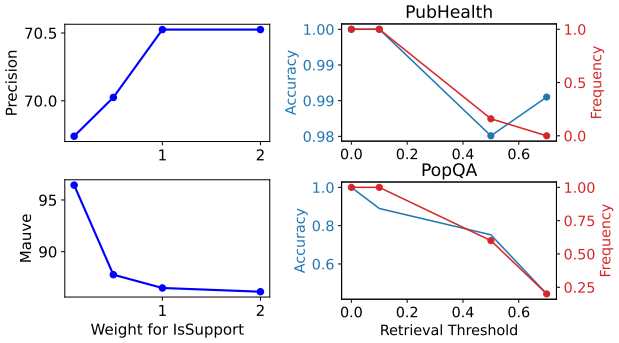
5.1 MAIN RESULTS

Comparison against baselines without retrieval. Table 2 (top) presents the baselines without retrieval. Our SELF-RAG (bottom two rows) demonstrates a substantial performance advantage over supervised fine-tuned LLMs in all tasks and even outperforms ChatGPT in PubHealth, PopQA, biography generations, and ASQA (Rouge and MAUVE). Our approach also significantly outperforms a concurrent method that employs sophisticated prompt engineering; specifically, on the bio generation task, our 7B and 13B models outperform the concurrent CoVE (Dhuliawala et al., 2023), which iteratively prompts Llama2$_{65B}$ to refine output.

Comparison against baselines with retrieval. As shown in Tables 2 (bottom), our SELF-RAG also outperforms existing RAG in many tasks, obtaining the best performance among non-proprietary LM-based models on all tasks. While our method outperforms other baselines, on PopQA or Bio, powerful instruction-tuned LMs with retrieval (e.g., LLama2-chat, Alpaca) show large gains from their non-retrieval baselines. However, we found that these baselines provide limited solutions for tasks where we cannot simply copy or extract sub-strings of retrieved passages. On PubHealth and ARC-Challenge, baselines with retrieval do not improve performance notably from their no-retrieval counterparts. We also observe that most baselines with retrieval struggle to improve citation accuracy. On ASQA, our model shows significantly higher citation precision and recall than all models except ChatGPT. Gao et al. (2023) found that ChatGPT consistently exhibits superior efficacy in this particular task, surpassing smaller LMs. Our SELF-RAG bridges this performance gap, even outperforming ChatGPT in citation precision, which measures whether the model-generated claim is fully supported by cited evidence. We also found that on the metrics for factual precision, SELF-RAG 7B occasionally outperforms our 13B due to the tendency of smaller SELF-RAG to often generate

	PQA (acc)	Med (acc)	AS (em)
SELF-RAG (50k)	45.5	73.5	32.1
Training			
No Retriever $\mathcal{R}$	43.6	67.8	31.0
No Critic $\mathcal{C}$	42.6	72.0	18.1
Test			
No retrieval	24.7	73.0	–
Hard constraints	28.3	72.6	–
Retrieve top1	41.8	73.1	28.6
Remove IsSup	44.1	73.2	30.6

(a) Ablation



(b) Customization (c) Retrieval

Figure 3: **Analysis on SELF-RAG:** (a) **Ablation studies** for key components of SELF-RAG training and inference based on our 7B model. (b) **Effects of soft weights** on ASQA citation precision and Mauve (fluency). (c) **Retrieval frequency** and *normalized* accuracy on PubHealth and PopQA.

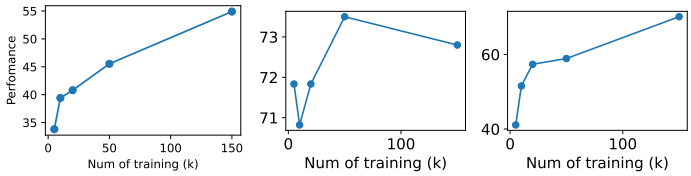
precisely grounded yet shorter outputs. Llama2-FT$_{7B}$, which is the baseline LM trained on the same instruction-output pairs as SELF-RAG without retrieval or self-reflection and is retrieval-augmented at test time only, lags behind SELF-RAG. This result indicates SELF-RAG gains are not solely from training data and demonstrate the effectiveness of SELF-RAG framework.

5.2 ANALYSIS

Ablation studies. We conduct a set of ablations of our framework to identify which factors play key roles. We evaluate two model variants trained differently than our model: *No Retriever* trains an LM using the standard instruction-following method given instruction-output pairs, without retrieved passages; *No Critic* trains an LM trained with input-output pairs that are always augmented with the top one retrieved document without reflection tokens. This is similar to SAIL (Luo et al., 2023), and we use our instruction-output data instead of using the Alpaca dataset (Dubois et al., 2023), as in SAIL. We also conduct ablation on our inference-time algorithm, including *No retrieval* disables retrieval during inference; *Hard constraints* indicates the model performance that retrieves when Retrieve=Yes instead of using the adaptive threshold; *Retrieve top 1* always retrieves and uses the top one document only, similar to standard RAG approaches; *Remove* IsSup indicates the model performance that removes IsSup score only during critique-guided beam search in Eq. 4. In this ablation experiment, we use a training instance size of 50k for a more efficient exploration of training variations. Later in this section, we conduct an analysis of the effect of training data size. We conduct the ablation studies on three datasets, PopQA, PubHealth, and ASQA. On ASQA, we evaluate models on sampled 150 instances and exclude ablations involving adaptive or no retrieval processes.

We show in Table 3a the ablation results. The top part of the table shows results for training ablations, and the bottom part is for inference ablations. We see that all components play important roles. We also observe a large performance gap between SELF-RAG and No Retriever or Critic baselines across tasks, indicating that training an LM with those models largely contributes to the performance gain of SELF-RAG. Using the top passages regardless of their relevance (Retrieve top 1) as in conventional RAG approaches causes a large drop in PopQA and ASQA, and removing IsSup during the beam search results hurts performance on ASQA. This demonstrates the effectiveness of SELF-RAG's capabilities of carefully selecting generations based fine-grained multiple criterion, instead of naively using all of the top passages from the retrieval model or solely depending on relevance scores.

Effects of inference-time customization. One key benefit of our proposed framework is that it enables us to control how much each critique type affects the final generation sampling. We analyze the effects of different parameter weights on the top of our 7B model during inference time on ASQA, where multiple evaluation aspects are considered. Figure 3b shows the effects of changing the weighting term for IsSup, which criticizes how supported the output is by the text passage. As the figure shows, increasing the weight leads to positive effects on the models' citation precision since this puts more emphasis on whether model generation is supported by the evidence. On the



(a) PopQA (b) PubHealth (c) ASQA (prec)

	Pop	Bio.
S & P	92.5	70.0
IsRel	95.0	90.0
IsSup	90.0	85.0

(d) Human evaluation on PopQA and Bio generation.

Figure 4: **Training scale and Human analysis:** (a) (b) (c) **Training scale analysis** shows the effect of the training data scale on PopQA, PubHealth and ASQA (citation precision), respectively. (d) **Human analysis** on SELF-RAG outputs as well as reflection tokens.

contrary, a larger weight results in lower MAUVE scores: when generation gets longer and more fluent, there are often more claims that are not fully supported by citations, consistent with findings by Liu et al. (2023a). Our framework lets practitioners choose and customize models' behaviors at test time by adjusting such parameters without requiring additional training.

Efficiency and accuracy trade-off. Using our framework, practitioners can adjust how often retrieval occurs using the token probability of reward tokens. We evaluate how this adaptive threshold affects overall accuracy and frequency of retrieval, and we evaluate the performance with varying numbers of threshold δ (larger δ results in less retrieval) on PubHealth and PopQA. Figure 3c shows that the model's retrieval frequencies dramatically change on both datasets. as δ varies. On one hand, performance deterioration by retrieving less is smaller on PubHealth but larger in PopQA.

Effects of training data size. We conduct an analysis of how the data scale affects the model's performance. In particular, we randomly sample 5k, 10k, 20k, and 50k instances from our original 150k training instances, and fine-tune four SELF-RAG $_{7B}$ variants on those subsets. Then, we compare the model performance on PopQA, PubHealth, and ASQA (citation precision) with our final SELF-RAG trained on the full 150k instances. We also evaluate Figures 4a, 4b and 4c shows the models' performance trained on different amount of data. Across all datasets, increasing data size often shows upward trajectories and the improvements are significantly larger in PopQA and ASQA, while we do not observed such significant improvements on Llama2-FT$_{7B}$ when increasing the training data from 50k to 150k. These results also indicate that further expanding the training data of SELF-RAG may lead to further improvements, although in this work we limit our training data size to 150k.

Human evaluations. We conduct small human evaluations on SELF-RAG outputs, as well as the reliability of predicted reflection tokens. In particular, we sampled 50 samples from PopQA and Bio results. Following Menick et al. (2022), human annotators evaluate *S&P*, which indicates whether the model output is plausible (i.e., the output is a reasonable and on-topic response to the question as if it were occurring in a conversation) and supported (i.e., the provided evidence is sufficient to verify the validity of the answer). For S&P, we do not consider the instances where SELF-RAG predicts `irrelevant` or `no support`. We then ask our annotators whether the model-predicted reflection tokens about IsRel and IsSup match their inspections (e.g., whether the *fully supported* output is supported by the cited evidence). Human annotators find SELF-RAG answers are often plausible and supported by relevant passages with higher S&P scores on short-form PopQA, which is consistent with Menick et al. (2022). Human annotators also find IsRel and IsSup reflection token predictions are mostly aligned with their assessments. Appendix Table 6 shows several annotated examples and explanations on assessments.

6 CONCLUSION

This work introduces SELF-RAG, a new framework to enhance the quality and factuality of LLMs through retrieval on demand and self-reflection. SELF-RAG trains an LM to learn to retrieve, generate, and critique text passages and its own generation by predicting the next tokens from its original vocabulary as well as newly added special tokens, called reflection tokens. SELF-RAG further enables the tailoring of LM behaviors at test time by leveraging reflection tokens. Our holistic evaluations on six tasks using multiple metrics demonstrate that SELF-RAG significantly outperforms LLMs with more parameters or with conventional retrieval-augmented generation approaches.

ETHICAL CONCERNS

This work aims to improve the factuality of LLM outputs, the lack of which continues to cause numerous real-world problems (e.g., spread of misinformation and provision of incorrect and dangerous advice). While our method shows significant improvements in terms of performance, factuality, and citation accuracy, it can still generate outputs that are not fully supported by the citations. We hope that explicit self-reflection and fine-grained attribution may help users verify factual errors in the model outputs.

ACKNOWLEDGMENTS

We thank Sewon Min, Scott Wen-tau Yih, Sean Welleck, and Kawin Ethayarajh for fruitful discussions in the early stages of this work. We thank Sewon Min, Joongwon (Daniel) Kim, and Sandy Kaplan for valuable feedback on the paper, and Tianyu Gao and Weijia Shi for their help on evaluations. Akari Asai is supported by the IBM Fellowship. We thank Stability AI for providing computing to train and evaluate the LMs in this work, and Microsoft Accelerate Foundation Models Research Program for the access to OpenAI APIs. This work was funded in part by the DARPA MCS program through NIWC Pacific (N66001-19-2-4031), NSF IIS-2044660, and gifts from AI2.

REFERENCES

Akari Asai, Kazuma Hashimoto, Hannaneh Hajishirzi, Richard Socher, and Caiming Xiong. Learning to retrieve reasoning paths over wikipedia graph for question answering. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SJgVHkrYDH.

Akari Asai, Sewon Min, Zexuan Zhong, and Danqi Chen. Retrieval-based language models and applications. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Tutorial)*, 2023a. URL https://aclanthology.org/2023.acl-tutorials.6.

Akari Asai, Timo Schick, Patrick Lewis, Xilun Chen, Gautier Izacard, Sebastian Riedel, Hannaneh Hajishirzi, and Wen-tau Yih. Task-aware retrieval with instructions. In *Findings of the Association for Computational Linguistics*, 2023b. URL https://aclanthology.org/2023.findings-acl.225.

Bernd Bohnet, Vinh Q Tran, Pat Verga, Roee Aharoni, Daniel Andor, Livio Baldini Soares, Jacob Eisenstein, Kuzman Ganchev, Jonathan Herzig, Kai Hui, et al. Attributed question answering: Evaluation and modeling for attributed large language models. *arXiv preprint arXiv:2212.08037*, 2022. URL https://arxiv.org/abs/2212.08037.

Lingjiao Chen, Matei Zaharia, and James Zou. How is chatgpt's behavior changing over time? *arXiv preprint arXiv:2307.09009*, 2023. URL https://arxiv.org/abs/2307.09009.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018. URL https://arxiv.org/abs/1803.05457.

Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. In *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=H4DqfPSibmx.

Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. Chain-of-verification reduces hallucination in large language models. *arXiv preprint arXiv:2309.11495*, 2023. URL https://arxiv.org/abs/2309.11495.

Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. Wizard of wikipedia: Knowledge-powered conversational agents. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=r1l73iRqKm.

Yann Dubois, Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Alpacafarm: A simulation framework for methods that

learn from human feedback. *arXiv preprint arXiv:2305.14387*, 2023. URL https://arxiv.org/abs/2305.14387.

Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. Enabling large language models to generate text with citations. *arXiv preprint arXiv:2305.14627*, 2023. URL https://arxiv.org/abs/2305.14627.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. Retrieval augmented language model pre-training. In *International Conference on Machine Learning*, 2020. URL https://dl.acm.org/doi/pdf/10.5555/3524938.3525306.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. Unsupervised dense information retrieval with contrastive learning. *Transactions on Machine Learning Research*, 2022a. URL https://openreview.net/forum?id=jKN1pXi7b0.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. Few-shot learning with retrieval augmented language models. *arXiv preprint arXiv:2208.03299*, 2022b. URL https://arxiv.org/abs/2208.03299.

Zhengbao Jiang, Frank F Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. Active retrieval augmented generation. *arXiv preprint arXiv:2305.06983*, 2023. URL https://arxiv.org/abs/2305.06983.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2017. URL https://aclanthology.org/P17-1147.

Nitish Shirish Keskar, Bryan McCann, Lav R Varshney, Caiming Xiong, and Richard Socher. Ctrl: A conditional transformer language model for controllable generation. *arXiv preprint arXiv:1909.05858*, 2019. URL https://arxiv.org/abs/1909.05858.

Tomasz Korbak, Kejian Shi, Angelica Chen, Rasika Vinayak Bhalerao, Christopher Buckley, Jason Phang, Samuel R Bowman, and Ethan Perez. Pretraining language models with human preferences. In *International Conference on Machine Learning*, 2023. URL https://openreview.net/forum?id=AT8Iw8KOeC.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 2019. URL https://aclanthology.org/Q19-1026.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023. URL https://arxiv.org/abs/2309.06180.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, 2020. URL https://proceedings.neurips.cc/paper/2020/file/6b493230205f780e1bc26945df7481e5-Paper.pdf.

Xi Victoria Lin, Xilun Chen, Mingda Chen, Weijia Shi, Maria Lomeli, Rich James, Pedro Rodriguez, Jacob Kahn, Gergely Szilvasy, Mike Lewis, Luke Zettlemoyer, and Scott Yih. Ra-dit: Retrieval-augmented dual instruction tuning, 2023. URL https://arxiv.org/abs/2310.01352.

Nelson F Liu, Tianyi Zhang, and Percy Liang. Evaluating verifiability in generative search engines. *arXiv preprint arXiv:2304.09848*, 2023a. URL https://arxiv.org/abs/2304.09848.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. Gpteval: Nlg evaluation using gpt-4 with better human alignment. *arXiv preprint arXiv:2303.16634*, 2023b. URL https://arxiv.org/abs/2303.16634.

Ximing Lu, Sean Welleck, Jack Hessel, Liwei Jiang, Lianhui Qin, Peter West, Prithviraj Ammanabrolu, and Yejin Choi. QUARK: Controllable text generation with reinforced unlearning. In *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=5HaIds3ux5O.

Hongyin Luo, Yung-Sung Chuang, Yuan Gong, Tianhua Zhang, Yoon Kim, Xixin Wu, Danny Fox, Helen Meng, and James Glass. Sail: Search-augmented instruction learning. *arXiv preprint arXiv:2305.15225*, 2023. URL https://arxiv.org/abs/2305.15225.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-refine: Iterative refinement with self-feedback. *arXiv preprint arXiv:2303.17651*, 2023. URL https://arxiv.org/abs/2303.17651.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 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)*, 2023. URL https://aclanthology.org/2023.acl-long.546.

Jacob Menick, Maja Trebacz, Vladimir Mikulik, John Aslanides, Francis Song, Martin Chadwick, Mia Glaese, Susannah Young, Lucy Campbell-Gillingham, Geoffrey Irving, et al. Teaching language models to support answers with verified quotes. *arXiv preprint arXiv:2203.11147*, 2022. URL https://arxiv.org/abs/2203.11147.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018. URL https://aclanthology.org/D18-1260.

Sewon Min, Danqi Chen, Hannaneh Hajishirzi, and Luke Zettlemoyer. A discrete hard EM approach for weakly supervised question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019. URL https://aclanthology.org/D19-1284.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. Factscore: Fine-grained atomic evaluation of factual precision in long form text generation. *arXiv preprint arXiv:2305.14251*, 2023. URL https://arxiv.org/abs/2305.14251.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*, 2021. URL https://arxiv.org/abs/2112.09332.

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. Large dual encoders are generalizable retrievers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 2022. URL https://aclanthology.org/2022.emnlp-main.669.

OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. URL https://arxiv.org/abs/2303.08774.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Gray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and

Ryan Lowe. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=TG8KACxEON.

Debjit Paul, Mete Ismayilzada, Maxime Peyrard, Beatriz Borges, Antoine Bosselut, Robert West, and Boi Faltings. Refiner: Reasoning feedback on intermediate representations. *arXiv preprint arXiv:2304.01904*, 2023. URL https://arxiv.org/abs/2304.01904.

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, Vassilis Plachouras, Tim Rocktäschel, and Sebastian Riedel. KILT: a benchmark for knowledge intensive language tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2021. URL https://aclanthology.org/2021.naacl-main.200.

Krishna Pillutla, Swabha Swayamdipta, Rowan Zellers, John Thickstun, Sean Welleck, Yejin Choi, and Zaid Harchaoui. MAUVE: Measuring the gap between neural text and human text using divergence frontiers. In *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=Tqx7nJp7PR.

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. Zero: Memory optimizations toward training trillion parameter models. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, 2020. URL https://dl.acm.org/doi/10.5555/3433701.3433727.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. In-context retrieval-augmented language models. *Transactions of the Association for Computational Linguistics*, 2023. URL https://arxiv.org/abs/2302.00083.

Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M Rush. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=9Vrb9D0WI4.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. *arXiv preprint arXiv:2302.04761*, 2023. URL https://arxiv.org/abs/2302.04761.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. URL https://arxiv.org/abs/1707.06347.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H. Chi, Nathanael Schärli, and Denny Zhou. Large language models can be easily distracted by irrelevant context. In *Proceedings of the 40th International Conference on Machine Learning*, 2023. URL https://proceedings.mlr.press/v202/shi23a.html.

Ivan Stelmakh, Yi Luan, Bhuwan Dhingra, and Ming-Wei Chang. ASQA: Factoid questions meet long-form answers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 2022. URL https://aclanthology.org/2022.emnlp-main.566.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. FEVER: a large-scale dataset for fact extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2018. URL https://aclanthology.org/N18-1074.

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

Yizhong Wang, Hamish Ivison, Pradeep Dasigi, Jack Hessel, Tushar Khot, Khyathi Raghavi Chandu, David Wadden, Kelsey MacMillan, Noah A Smith, Iz Beltagy, et al. How far can camels go? exploring the state of instruction tuning on open resources. *arXiv preprint arXiv:2306.04751*, 2023. URL https://arxiv.org/abs/2306.04751.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=gEZrGCozdqR.

Zeqiu Wu, Yushi Hu, Weijia Shi, Nouha Dziri, Alane Suhr, Prithviraj Ammanabrolu, Noah A Smith, Mari Ostendorf, and Hannaneh Hajishirzi. Fine-grained human feedback gives better rewards for language model training. *arXiv preprint arXiv:2306.01693*, 2023. URL https://arxiv.org/abs/2306.01693.

Yuxi Xie, Kenji Kawaguchi, Yiran Zhao, Xu Zhao, Min-Yen Kan, Junxian He, and Qizhe Xie. Decomposition enhances reasoning via self-evaluation guided decoding. *arXiv preprint arXiv:2305.00633*, 2023. URL https://arxiv.org/abs/2305.00633.

Fangyuan Xu, Weijia Shi, and Eunsol Choi. Recomp: Improving retrieval-augmented lms with compression and selective augmentation, 2023. URL https://arxiv.org/abs/2310.04408.

Ori Yoran, Tomer Wolfson, Ori Ram, and Jonathan Berant. Making retrieval-augmented language models robust to irrelevant context, 2023. URL https://arxiv.org/abs/2310.01558.

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

Tianhua Zhang, Hongyin Luo, Yung-Sung Chuang, Wei Fang, Luc Gaitskell, Thomas Hartvigsen, Xixin Wu, Danny Fox, Helen Meng, and James Glass. Interpretable unified language checking. *arXiv preprint arXiv:2304.03728*, 2023. URL https://arxiv.org/abs/2304.03728.

Andy Zhou, Kai Yan, Michal Shlapentokh-Rothman, Haohan Wang, and Yu-Xiong Wang. Language agent tree search unifies reasoning acting and planning in language models, 2023. URL https://arxiv.org/abs/2310.04406.

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

APPENDIX

A SELF-RAG Details **17**
A.1 Reflection Tokens. 17
A.2 SELF-RAG Training 17
A.3 SELF-RAG Inference 19

B Experimental Details **19**
B.1 More Details of Training 19
B.2 More Details of Evaluations 20

C Results **20**
C.1 Analysis 20
C.2 Human Evaluation Examples 21
C.3 Qualitative Examples 21

D Full List of Instructions and Demonstrations for GPT-4 **21**

A SELF-RAG DETAILS

A.1 REFLECTION TOKENS.

Definitions of reflection tokens. Below, we provide a detailed definition of reflection type and output tokens. The first three aspects will be provided at each segment level, while the final aspect is only given at each output level.

- **Retrieval-on-demand** (Retrieve): Given an input and previous-step generation (if applicable), an LM determines whether the continuation requires factual grounding. `No` indicates retrieval is unnecessary as the sequence does not require factual grounding or may not be enhanced by knowledge retrieval, `Yes` indicates retrieval is necessary. We additionally have `continue to use evidence`, which indicates that a model can continue to use the evidence retrieved previously. For instance, a passage may contain rich factual information, and thus SELF-RAG generates multiple segments based on the passage.
- **Relevant** (IsRel): Retrieved knowledge may not be always relevant to the input. This aspect indicates whether the evidence provides useful information (`Relevant`) or not (`Irrelevant`).
- **Supported** (IsSup): Attribution is the concept of whether the output is fully supported by certain evidence (Menick et al., 2022; Bohnet et al., 2022). This aspect judges how much information in the output is entailed by the evidence. We evaluate attributions in three scale, `Fully supported`, `Partially supported`, and `No support / Contradictory`, following Yue et al. (2023); Nakano et al. (2021).
- **Useful** (IsUse): Following the definitions from Liu et al. (2023a), we define the perceived utility as whether the response is a helpful and informative answer to the query, independently from whether it is in fact factual or not. This can be also viewed as plausibility in Menick et al. (2022). For usefulness, we use a five-scale evaluation (1 is the lowest and 5 is the highest).

Details of GPT-4-based data collections. We use the instruction and demonstration pairs to prompt GPT-4, listed in Section D. Following an official recommendation, we separate instructions and outputs with "##". We use the temperature 1 and set the maximum output token counts to be 200. We discard instances where GPT-4 does not follow the designated output formats or output sequences that do not match our expected category names. As a result, we collected 1,2594 for Retrieve, 11,181 for IsSup , 19,317 for relevance, 3,831 for utility.

Manual analysis of the GPT-4 predictions. The authors of this paper manually assess randomly sampled 20 instances for each aspect and check if GPT-4 predictions match their assessments given the same instruction, demonstrations, and test instances. We found our assessments show high agreement with GPT-4 predictions, especially for relevance (95%), retrieval necessity (95%), and the degree of support (90%). Agreement was slightly lower in usefulness (80%), mostly due to the disagreement between 1 and 2 or 4 and 5.

A.2 SELF-RAG TRAINING

Overview of training. Algorithm 2 provides a high-level overview of our training.

Full list of seed datasets. To sample diverse input-output pairs, we sample instances of the OpenInstruct (Wang et al., 2023) dataset. In particular, we use their ShareGPT, GPT-4 Alpaca, Alpaca, OpenAssistant, and FLAN subsets subsets. We also sample instances from a couple of knowledge-intensive datasets, Natural Questions (Kwiatkowski et al., 2019), Wizard of Wikipedia (Dinan et al., 2019) and FEVER (Thorne et al., 2018) from the KILT benchmark (Petroni et al., 2021), ASQA (Stelmakh et al., 2022) and multiple QA datasets including ARC-Easy and OpenBookQA (Mihaylov et al., 2018). Table 3 shows the full list of training instances, and in total, we use 145,619 instances.

Performance of the Critic $\mathcal{C}$. We evaluate the accuracy of reward predictions by splitting GPT-4 generated feedback into training, development, and test sets. The accuracy of the reward model is as follows. Table 5 shows the model performance of predicting GPT-4 judgments. As you can see, overall our fine-tuned reward model shows high prediction matching with GPT-4 predicted feedback.

Algorithm 2 SELF-RAG Training

Input input-output data $\mathcal{D} = \{X, Y\}$, generator $\mathcal{M}$, $\mathcal{C}$ θ
Initialize $\mathcal{C}$ with a pre-trained LM
Sample data $\{X^{sample}, Y^{sample}\} \sim \{X, Y\}$ ▷ **Training Critic LM (Section 3.2.1)**
for $(x, y) \in (X^{sample}, Y^{sample})$ **do** ▷ Data collections for $\mathcal{C}$
    Prompt GPT-4 to collect a reflection token r for (x, y)
    Add $\{(x, y, r)\}$ to $\mathcal{D}_{critic}$
Update $\mathcal{C}$ with next token prediction loss ▷ Critic learning; Eq. 1
Initialize $\mathcal{M}$ with a pre-trained LM ▷ **Training Generator LM (Section 3.2.2)**
for $(x, y) \in (X, Y)$ **do** ▷ Data collection for $\mathcal{M}$ with $\mathcal{D}_{critic}$
    Run $\mathcal{C}$ to predict r given (x, y)
    Add (x, y, r) to $\mathcal{D}_{gen}$
Update $\mathcal{M}$ on $\mathcal{D}_{gen}$ with next token prediction loss ▷ Generator LM learning; Eq. 2

Dataset name	category	Data source	the number of instances
GPT-4 Alpaca	Instruction-following	Open-Instruct	26,168
Stanford Alpaca	Instruction-following	Open-Instruct	25,153
FLAN-V2	Instruction-following	Open-Instruct	17,817
ShareGPT	Instruction-following	Open-Instruct	13,406
Open Assistant 1	Instruction-following	Open-Instruct	9,464
Wizard of Wikipedia	Knowledge-intensive	KILT	17,367
Natural Questions	Knowledge-intensive	KILT	15,535
FEVER	Knowledge-intensive	KILT	9,966
OpenBoookQA	Knowledge-intensive	HF Dataset	4,699
Arc-Easy	Knowledge-intensive	HF Dataset	2,147
ASQA	Knowledge-intensive	ASQA	3,897

Table 3: The generator LM $\mathcal{M}$ training data statistics.

base LM	Retrieve	IsSup	IsRel	IsUse
Llama2-7B	**93.8**	**93.5**	80.2	**73.5**
FLAN-3B	85.6	73.1	**82.0**	72.1

Figure 5: Reward prediction accuracy using GPT-4 predictions as ground-truth predictions.

While our final model uses Llama2-7B as a base LM, we also train and compare FLAN-3B (Wei et al., 2022) model on the same data, to investigate the effectiveness of different data sizes affect final reward predictions. In most aspects, our reward model shows higher than 80% accuracy, indicating the powerful ability of fine-tuned specialized LMs to evaluate text. While both models show relatively lower performance on IsUse, this is because both models often confuse between the two highest cases (5 and 4), where human annotators can also disagree.

Details of $\mathcal{M}$ data creation. Here, we provide detailed data creation procedures. Algorithm 3 summarizes the process. Here we set y_t to y for simplification. Once we train the critic model, we first run it on input data from the aforementioned datasets, to predict whether retrieval is needed or not. For the instances where the critic predicts Retrieve=No, we only predict the IsUse given input and output. For the instances where the critic predicts Retrieve=Yes, we first retrieve passages using the input and the entire output as queries, to find passages that are relevant to the entire output. We then split output sentences using Spacy.[7] For each sentence, we run $\mathcal{C}$ to predict whether the retrieval is necessary or not, given the input, preceding segments, and the initial retrieved passage. If $\mathcal{C}$ predicts Retrieve=No, then do not insert any paragraph at the tth segment. If $\mathcal{C}$ predicts Retrieve=Yes, then we use the original input and the tth segment as a retrieval query to find relevant passages for the t-th segment. For each retrieved passage, we predict IsRel and IsSup. If there is any passage and continuation with IsRel =Relevant and IsSup =Fully Supported/ IsSup =Partially

[7] https://spacy.io/

`Supported`, then we sample it as the continuation. If there is more than one passage satisfying this criterion, we use the one with the highest retrieval score. If there are only [IsRel] =`Irrelevant` or [IsSup] =`No Support` passages, we randomly sample one passage.

Algorithm 3 $\mathcal{M}_{gen}$ Data creation

Input Input-output data $\mathcal{D} = X, Y$
for $(x, y) \in \{X, Y\}$ **do**
  Given (x, y) $\mathcal{C}$ predicts [Retrieve]
  if [Retrieve] is predicted **then**
    Retrieve relevant passages $\mathbf{D}$ using $\mathcal{R}$ given (x, y) ▷ Retrieve passages
    for $d \in \mathbf{D}$ **do**
      $\mathcal{C}$ predicts [IsRel] for each d ▷ Predict relevance of passages
      $\mathcal{C}$ predicts [IsSup] for each (y, d) ▷ Predict supports of outputs
    $\mathcal{C}$ predicts [IsUse] for each d ▷ Predict overall utility ($t = \mathcal{T}$ only)
    Sample d
  else if [Retrieve] is not predicted **then**
    $\mathcal{C}$ predicts [IsUse] given x, y
  Add augmented (x, y, d, r) to $\mathcal{D}_{gen}$

Training examples. Table 4 show several training examples used for $\mathcal{M}$ training.

A.3 Self-Rag Inference

Details of beam-search score calculations. We first compute scores for each critique type by taking the normalized probabilities of desirable tokens. For [IsRel], we compute the score as follows:

$$s(\boxed{\text{IsRel}}) = \frac{p(\boxed{\text{IsRel}} = \text{Relevant})}{p(\boxed{\text{IsRel}} = \text{Relevant}) + p(\boxed{\text{IsRel}} = \text{Irrelevant})}.$$

For [IsSup], we compute the score as follows:

$$s(\boxed{\text{IsRel}}) = \frac{p(\boxed{\text{IsSup}} = \text{Fully})}{S} + 0.5 \times \frac{p(\boxed{\text{IsSup}} = \text{Partially})}{S},$$

where $S = \sum_{t \in \{\text{Fully,Partially,No}\}} p(\boxed{\text{IsSup}} = t)$. For [IsUse] where we have a five-scale score, we compute the weighted sum of the scores. We assigns weighted scores of $w = \{-1, -0.5, 0, 0.5, 1\}$ to the tokens [IsUse] $=\{1, 2, 3, 4, 5\}$, and compute the final scores as follows:

$$s(\boxed{\text{IsUse}}) = \sum_{i}^{5} w_i \frac{p(\boxed{\text{IsUse}} = i)}{S},$$

where $S = \sum_{t \in \{1,2,3,4,5\}} p(\boxed{\text{IsUse}} = t)$.

Details of adaptive retrieval. For retrieval based on soft constraints, we trigger retrieval if the following condition is satisfied:

$$\frac{p(\boxed{\text{Retrieve}} = \text{Yes})}{p(\boxed{\text{Retrieve}} = \text{Yes}) + p(p(\boxed{\text{Retrieve}} = \text{No})} > \delta.$$

B Experimental Details

B.1 More Details of Training

More details of training and computations. We use 4 Nvidia A100 with 80GB memory to train our models. All models are trained for 3 epochs with a batch size of 128, a peak learning rate of 2e-5 with 3% warmup steps, and linear decay afterward. We set the maximum token length to be 2,048 for the 7B model, and 1,524 for the 13B model due to the memory constraint. We use Deepspeed stage 3 (Rajbhandari et al., 2020) to conduct multi-GPU distributed training, with training precision Bfloat16 enabled. FlashAttention (Dao et al., 2022) is used to make the long-context training more efficient. We run inference of our trained models using 1-2 Quadro RTX 6000 GPUs with 24GB memory.

B.2 More Details of Evaluations

Retrieval setup details. By default, we use Contriever-MS MARCO to retrieve the top five documents from Wikipedia, and use official Wikipedia embeddings based on 2018 English Wikipedia. On PopQA, where question and answer pairs are created based on WikiData in 2022, we found that the 2018 Wikipedia sometimes lacks articles about some entities that have been more recently added to Wikipedia. Therefore, for PopQA, we used the December 2020 preprocessed Wikipedia corpus provided by Izacard et al. (2022b) and generated document embeddings.[8] The issues of performance variance from different Wikipedia dumps have been reported by prior work (Asai et al., 2020; Izacard et al., 2022b). Yet, we observe limited effectiveness of such off-the-shelf retrieval models trained primarily on knowledge-intensive tasks for open-ended generation (e.g., instruction following). Recent or concurrent work studies instruction-tuning of retrieval systems (Asai et al., 2023b) or joint training of retrieval and LM components (Lin et al., 2023), while we leave exploring the effectivess of such appraoches for future work. For bio generation and open-domain QA tasks, we additionally retrieve five documents using Google Programmable Search[9] and search documents from English Wikipedia. As this API only provides snippets, we retrieve Wikipedia introductory paragraphs for the corresponding entities.

Detailed experimental settings for individual datasets. For OpenQA datasets, we set the maximum new token number to 100 tokens. For closed-set tasks (PubHealth and ARC-C), we set the maximum new token length to 50 for all baselines. For Self-Rag inference on PubHealth and ARC-C, instead of determining the output with the highest score 4 as in other tasks, we aggregate the scores for each option and select the answer option with the highest score. We found in zero-shot settings of fact checking, some LLMs can generate capitalized class labels (e.g., True) while our gold labels are lower-cased. Therefore, across different LMs, for fact checking, we lowercase the predictions. In multiple choice tasks, we found some models generate answers in slightly different ways (e.g., (A) instead of A). We slightly modify instructions for each LLM to avoid such format violations, and further conduct string matching between each candidate and model predictions if format violations still remain. After that processing, in closed set tasks, model predictions match one of the gold classes in almost all cases. For ALCE, we found that Llama2-chat tend to generate significantly lower outputs than other models (e.g., on average, their output is nearly 100 token, while ChatGPT generates 40 tokens on average), resulting in inflated str-em scores. We limit the maximum generation length to 100 tokens for all baselines to avoid this issue, rather than the original 300 tokens in the ALCE paper. Consequently, all of the baseline output length is within 30-60 tokens. For FactScore, we set the maximum new token length to 500 for baselines and 200 for Self-Rag at each segment level.

Task-specific instructions. Table 5 shows the list of the instructions used during evaluations. For Open-domain QA, we do not provide explicit instructions.

C Results

C.1 Analysis

Reliance on parametric- and non-parametric memories. We conduct analysis on how frequently model answers come from retrieved passages (non-parametric memories) or their own parametric memories. On two open-domain QA datasets, TriviaQA and PopQA, we conduct the following analysis: 1) sample query models successfully answer correctly, 2) for each query in this group, check whether the matched ground-truth answer is a sub-string of the retrieved passage or not. We evaluate Self-Rag 7B, Alpaca 7B, Alpaca 13B, and Llama2-Chat-13B. We found that Self-Rag significantly less frequently generates answers that are not included in the provided evidence; in particular, in Alpaca 30B, 20% of the correct predictions are not included in the provided passages, followed by Llama2-chat 13B (18%) and Alpaca (15%), while it is only 2% in Self-Rag. When retrieved passages are not relevant, Self-Rag generates IsRel =`Irrelevant`, indicating that the following answers may not be factually grounded, while those instruction-tuned models continue to generate plausible answers.

[8] `https://github.com/facebookresearch/atlas`

[9] `https://programmablesearchengine.google.com/about/`

C.2 Human Evaluation Examples

Table 6 shows examples with human evaluations on S&P and correctness of [IsRel] and [IsSup] reflection tokens.

C.3 Qualitative Examples

Table 7 shows several examples predicted by our Self-Rag (13B). The first example is the model output to an ASQA question. The first reference states that Emperor Constantine made Sunday a day of rest from labor, and further the second citation supports the fact that the official adoption of Sunday as a day of rest by Constantine in AD 321. In the second example, the model predicts `Contradictory` to the first output as the output says the person has served as the CEO since 2010, while the passage says he stepped down as CEO in 2015. Indicating those factual contradictions as reflection tokens enables to enforcement of hard control and also verification of model outputs easily. In the third example, while the generation is mostly correct, Self-Rag predicts `Partially Support` to the statement listing the name of the songs, as they were not explicitly mentioned.

D Full List of Instructions and Demonstrations for GPT-4

Here, we show the instructions and demonstrations used to prompt GPT-4 to collect reflection tokens. Table 8 shows the instructions and demonstrations for the initial retrieval token. Table 9 shows the instruction and demonstrations used to collect the three-way output tokens for [Retrieve] given instruction, preceding sentences, and previously retrieved passages. Due to the longer demonstration and test input, we only use a single demonstration. Table 10 shows an instruction and demonstrations used to collect the three-way output tokens for [IsRel] . Table 11 shows an instruction and demonstrations used to collect the three-way output tokens for [IsRel] . Table 12 shows an instruction and demonstrations used to collect the five-way output tokens for [IsUse] .

Input Identify the two types of computer memory. (from gpt4_alpaca)

Output [Retrieve=Yes]<p>Computer memory Computer memory – In computing, memory refers to the computer hardware integrated circuits that store information for immediate use in a computer; it is synonymous with the term "primary storage". Computer memory operates at a high speed, for example random-access memory (RAM), as a distinction from storage that provides slow-to-access information but offers higher capacities. If needed, contents of the computer memory can be transferred to secondary storage; a very common way of doing this is through a memory management technique called "virtual memory". An archaic synonym for memory is store. The term "memory", meaning "primary storage" or "main memory", is<p>[IsRel =Relevant]Primary memory, also known as main memory, is the memory that the computer can directly access and manipulate. [IsSup =Fully Supported] [Retrieve=Yes]<p>Computer memory–often associated with addressable semiconductor memory, i.e. integrated circuits consisting of silicon-based transistors, used for example as primary storage but also other purposes in computers and other digital electronic devices. There are two main kinds of semiconductor memory, volatile and non-volatile. Examples of non-volatile memory are flash memory (used as secondary memory) and ROM, PROM, EPROM and EEPROM memory (used for storing firmware such as BIOS). Examples of volatile memory are primary storage, which is typically dynamic random-access memory (DRAM), and fast CPU cache memory, which is typically static random-access memory (SRAM) that is fast but energy-consuming, offering lower memory<p>[IsRel =Relevant]Primary memory can be accessed quickly, but it is volatile, meaning that the data is lost if the power is turned off.[IsSup =Partially Supported]...[IsUse :5]

Input Give an example of a situation where you have to delegate tasks (from gpt4_alpaca)

Output [Retrieve=No]The manager realizes that there is an extensive list of tasks that need to be completed in order to make the event a success.[Retrieve=Yes]<p>Delegation–Delegation is the assignment of any responsibility or authority to another person (normally from a manager to a subordinate) to carry out specific activities. It is one of the core concepts of management leadership. However, the person who delegated the work remains accountable for the outcome of the delegated work. Delegation empowers a subordinate to make decisions, i.e. it is a shifting of decision-making authority from one organizational level to a lower one. Delegation, if properly done, is not fabrication. The opposite of effective delegation is micromanagement, where a manager provides too much input,<p>[IsRel =Relevant]This includes tasks such as creating a guest list, designing and sending invitations, booking a venue, coordinating with vendors, planning the eventś program, and monitoring the budget.[IsSup =Partially Supported][Retrieve=No]Since it is not possible for the manager to handle everything alone, they must delegate tasks to their team members.[Retrieve=No]The manager would then assign each team member specific responsibilities based on their individual skillsets and strengths.[Retrieve=Yes]<p>Joint application design – Subject Matter Experts: These are the business users, the IS professionals, and the outside experts that will be needed for a successful workshop. This group is the backbone of the meeting<p>[IsRel =Irrelevant]For instance, the teamś graphic designer might be tasked with designing the invitations, while the event coordinator would be in charge of planning the program and working with vendors...[Retrieve=No]By delegating tasks, the manager can ensure that all necessary tasks are completed in a timely manner, while also empowering their team members to take on responsibilities and contribute to the success of the event.[IsUse =5]

Input Synthesize a poem based on this prompt:"The sun is setting" (from stanford_alpaca)

Output [Retrieve=No]The stars begin to twinkle bright the trees, scattered throughout the night; [Retrieve=No]The stars begin to twinkle bright the trees, scattered throughout the night [Retrieve=No] They flicker like a silent dance, A peaceful moment in its trance. [Retrieve=No] They flicker like a silent dance, A peaceful moment in its trance. [Retrieve=No] A chorus of crickets sing a lullaby, To every creature that roams the sky [IsUse =5]

Input when did walking dead season 7 come out (from nq)

Output [Retrieve=Yes]<p>The Walking Dead (season 7)–The Walking Dead (season 7) The seventh season of "The Walking Dead", an American post-apocalyptic horror television series on AMC, premiered on October 23, 2016, and concluded on April 2, 2017, consisting of 16 episodes. Developed for television by Frank Darabont, the series is based on the eponymous series of comic books by Robert Kirkman, Tony Moore, and Charlie Adlard. ...<p>[IsRel =Relevant]October 23 , 2016[IsSup =Fully Supported][IsUse =5]

Input Given a chat history separated by new lines, generates an informative, knowledgeable and engaging response: Hello, I love German shepherd dogs and you? – My favorite dog, they are officially known as German Shepherd Dogs? – Yes, I have a German shepherd in my house and she is pregnant (from wow)

Output hey also are easily trained, obedient, and intelligent.[Retrieve=Yes]<p>German Shepherd – a simple difference of opinion, it is the fundamental issue of the breedś essential conformation and movement. ... They are known for being easy to train and good for performing tasks and following instructions. They are especially well known for their police work, being used for tracking criminals, patrolling troubled areas and detecting and holding suspects. <p>[IsRel =Relevant]Because of this, they are often used for disability assistance and for police work. [IsSup =Partially Supported][IsUse =2]

Table 4: List of the training examples.

Dataset	Instruction
ARC-C	Given four answer candidates, A, B, C and D, choose the best answer choice. Please answer with the capitalized alphabet only, without adding any extra phrase or period.
PubHealth	Is the following statement correct or not? Say true if it's correct; otherwise, say false. Don't capitalize or add periods, just say "true" or "false".
Bio Generation	Tell me a bio about `[Person Name]`
ASQA (baseline)	Instruction: Write an accurate, engaging, and concise answer for the given question using only the provided search results (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.
ASQA (ours)	Answer the following question. The question may be ambiguous and have multiple correct answers, and in that case, you have to provide a long-form answer including all correct answers.

Table 5: Full list of instructions used during zero-shot evaluations. For open-domain QA, we don't use any task specific instruction and simply use the original questions as input query.

Input What genre is A Question and Answer Guide to Astronomy? (PopQA)

Output [Retrieve=Yes] <p>**Reference [1]**<p>[IsRel =Relevant] A Question and Answer Guide to Astronomy is a non-fiction book that falls under the genre of Astronomy and Cosmology. [IsSup =Fully Supported][IsUse =5]

Reference [1] A Question and Answer Guide to Astronomy is a book about astronomy and cosmology, and is intended for a general audience. The book was written by Pierre-Yves Bely, Carol Christian, and Jean-Rene Roy, and published in English by Cambridge University Press in 2010. It was originally written in French. The content within the book is written using a question and answer format. It contains some 250 questions, which The Science Teacher states each are answered with a concise and well-formulated essay that is informative and readable. The Science Teacher review goes on to state that many of the answers given in the book are little gems of science writing. The Science Teacher summarizes by stating that each question is likely to be thought of by a student, and that the answers are informative, well constructed, and thorough. The book covers information about the planets, the Earth, the Universe, practical astronomy, history, and awkward questions such as astronomy in the Bible, UFOs, and aliens. Also covered are subjects such as the Big Bang, comprehension of large numbers, and the Moon illusion.

S&P Supported and Plausible.

Correctness of IsRel **and** IsSup **:** Both IsRel and IsSup are correct, as the Reference [1] provides detailed information of the book including the genre, and the model output is fully supported by the reference.

Input Who is the author of The Lie? (PopQA)

Output [Retrieve=Yes] <p>**Reference [1]**<p>[IsRel =Relevant] The author of The Lie is Sam Harris.[IsSup =Fully Supported][IsUse =5]

Reference [1] Lying is a 2011 long-form essay book by American author and neuroscience expert Sam Harris. Harris argues that we can radically simplify our lives and improve society by merely telling the truth in situations where others often lie.

S&P Plausible but not supported.

Correctness of IsRel **and** IsSup Both IsRel and IsSup are incorrect, as the Reference [1] is about a different book whose title is similar to The Lie.

Input Tell me a bio about Ronaldinho (Bio)

Output [Retrieve=Yes] <p>**Reference [1]**<p>[IsRel =Relevant] Ronaldinho is a retired professional footballer from Brazil.[IsSup =Fully Supported][Retrieve=Continue]He played as an attacking midfielder and was known for his technical skills, creativity, dribbling ability, and accuracy from free kicks[Retrieve=Continue]He won two FIFA World Player of the Year awards and a Ballon d'Or.[IsUse =5]

Reference [1] Ronaldo de Assis Moreira (born 21 March 1980), commonly known as Ronaldinho Gaúcho or simply Ronaldinho, is a Brazilian retired professional footballer who played mostly as an attacking midfielder, but was also deployed as a winger. Considered one of the best players of his generation and regarded by many as one of the greatest of all time, Ronaldinho won two FIFA World Player of the Year awards and a Ballon d'Or. A global icon of the sport, he was renowned for his technical skills, creativity, dribbling ability and accuracy from free-kicks, as well as his use of tricks, feints, no-look passes and overhead...

S&P Plausible and fully supported.

Correctness of IsRel **and** IsSup Both IsRel and IsSup are correct.

Input Tell me a bio about Sergei Bodrov (Bio)

Output [Retrieve=Yes] <p>**Reference [1]**<p>[IsRel =Relevant] Sergei Vladimirovich Bodrov (born June 28, 1948) is a Russian film director, screenwriter, and producer.[IsSup =Fully Supported][Retrieve=Continue]He was born in Moscow, Russia.[Retrieve=Continue]He is the president of the Russian Guild of Film Producers.[IsUse =5]

Reference [1] Sergei Vladimirovich Bodrov (born June 28, 1948) is a Russian film director, screenwriter, and producer. In 2003 he was the President of the Jury at the 25th Moscow International Film Festival.

S&P Plausible but only partially supported. The the Reference [1] does not explicitly mention the person is born in Moscow and there is no description about the Russian Guild of Film Producers.

Correctness of IsRel **and** IsSup IsRel and IsSup are correct as the reference document is highly relevant and the first sentence is fully supported.

Table 6: Examples of human evaluation results.

Input When did sunday became the seventh day of the week (asqa)

Output [Retrieve=Yes] <p>**Reference [1]**<p>[IsRel =Relevant] In 321 AD, the Roman Emperor Constantine the Great made Sunday a day of rest from labor, stating: "On the venerable Day of the Sun let the magistrates and people residing in cities rest, and let all workshops be closed".[IsSup =Fully Supported] [Retrieve=Yes] <p>**Reference [2]**<p>[IsRel =Relevant] The official adoption of Sunday as a day of rest by Constantine the Great in AD 321 made Sunday the seventh day of the week.**[2]**[IsSup =Fully Supported][IsUse =5]

Reference [1] Emperor Constantine then enacted the first Sunday Laws, for "the venerable Day of the Sun" in 321 A.D. On March 7, 321, the Roman emperor Constantine issued a decree making Sunday a day of rest from labor, stating: We find a good description of this in sources such as the book Rest Days which states: Early Christian observance of both the spiritual seventh-day sabbath and a Lordś Day assembly is evidenced in Ignatiusś letter to the Magnesians ca. 110.

Reference [2] Despite the official adoption of Sunday as a day of rest by Constantine, the seven-day week and the nundial cycle continued to be used side-by-side until at least the Calendar of 354 and probably later. ... The fact that the canon had to be issued at all is an indication that adoption of Constantine's decree of 321 was still not universal ...

Input Tell me a bio about Rory Tapner. (bio generation)

Output <p>**Reference [1]**<p>[IsRel =Relevant] Rory Tapner is a British businessman who has served as the Chief Executive Officer of Coutts, a private banking and wealth management company, since 2010.[IsSup =Contradictory], **Reference [2]**<p>[IsRel =Relevant] Rory Tapner was born on 30 September 1959 in London, England.[IsSup =Fully Supported]

Reference [1] Rory Tapner Rory Tapner is a UK businessman who, in September 2010 was appointed CEO of Coutts, the Wealth Management division of Royal Bank of Scotland Group. In February 2015 he stepped down as CEO when he was replaced in the role by Alison Rose

Reference [2] Holdings (from August 2016); and the Council of the University of Buckingham (from July 2017). Rory Tapner Rory Tapner (born 30 September 1959) is a UK businessman ...

Input Tell me a bio about G. Venugopa. (bio generation)

Output (ranked 1) [Retrieve=Yes] <p>**Reference [1]**<p>[IsRel =Relevant] G. Venugopal is a popular playback singer in the Malayalam film industry. [Retrieve=Continue] He has sung over 300 songs in Malayalam, Tamil, and Hindi movies.[IsSup =Partially Supported] [Retrieve=Yes] <p>**Reference [2]**<p>G. Venugopal was born on December 10, 1960 in India.[IsSup =Fully Supported]

Reference [1] 'G. Venugopal G Venugopal (born 10 December 1960) is an Indian playback singer known for his work in Malayalam films. He started his singing career in the film "Odaruthammaava Aalariyaam" (1984). Since then he has sung in more than 300 films; and has over 500 private albums to his credit. The many awards he has won include the coveted Best Playback Singer (Kerala State) which he won thrice, the Film Critics award and the Kerala Kaumudi Gallop Poll award (twice). He won The Kerala State award for Best Playback singer (2004) third time for the song "aadedee..aadaadedee" from the film "Ullam".Venugopal'

Reference [2] Kerala State Film Awards: Kerala Film Critics Award Kerala State Government Award (Professional Drama): Asianet Film Awards: Kerala Kaumadi Gallup Poll Award: Academic G. Venugopal G Venugopal (born 10 December 1960) is an Indian playback singer known for his work in Malayalam films.

Table 7: Examples of outputs.

Instructions
Given an instruction, please make a judgment on whether finding some external documents from the web (e.g., Wikipedia) helps to generate a better response. Please answer [Yes] or [No] and write an explanation.

Demonstrations
Instruction Give three tips for staying healthy.
Need retrieval? [Yes]
Explanation There might be some online sources listing three tips for staying healthy or some reliable sources to explain the effects of different behaviors on health. So retrieving documents is helpful to improve the response to this query.

Instruction Describe a time when you had to make a difficult decision.
Need retrieval? [No]
Explanation This instruction is asking about some personal experience and thus it does not require one to find some external documents.

Instruction Write a short story in third person narration about a protagonist who has to make an important career decision.
Need retrieval? [No]
Explanation This instruction asks us to write a short story, which does not require external evidence to verify.

Instruction What is the capital of France?
Need retrieval? [Yes]
Explanation While the instruction simply asks us to answer the capital of France, which is a widely known fact, retrieving web documents for this question can still help.

Instruction Find the area of a circle given its radius. Radius = 4
Need retrieval? [No]
Explanation This is a math question and although we may be able to find some documents describing a formula, it is unlikely to find a document exactly mentioning the answer.

Instruction Arrange the words in the given sentence to form a grammatically correct sentence. quickly the brown fox jumped
Need retrieval? [No]
Explanation This task doesn't require any external evidence, as it is a simple grammatical question.

Instruction Explain the process of cellular respiration in plants.
Need retrieval? [Yes]
Explanation This instruction asks for a detailed description of a scientific concept, and is highly likely that we can find a reliable and useful document to support the response.

Table 8: Instructions and demonstrations for Retrieve aspect given the input only.

Instructions
You will be provided with an instruction, evidence, output sentence, and preceding sentences (optional). If the preceding sentence is given, the output should be the sentence that follows those preceding sentences. Your task is to determine whether the information in the output sentence can be fully verified by the evidence or if it requires further external verification. There are three cases:
- If the output sentence can be verified solely with the evidence, then respond with [Continue to Use Evidence].
- If the sentence doesn't require any factual verification (e.g., a subjective sentence or a sentence about common sense), then respond with [No Retrieval].
- If additional information is needed to verify the output sentence, respond with [Retrieval].
Please provide explanations for your judgments.

Instruction Explain the use of word embeddings in Natural Language Processing.
Preceding sentences Word embeddings are one of the most powerful tools available for Natural Language Processing (NLP). They are mathematical representations of words or phrases in a vector space, allowing similarities between words and the context in which they are used to be measured.
Evidence: Word embedding
Word embedding is the collective name for a set of language modeling and feature learning techniques in natural language processing (NLP) where words or phrases from the vocabulary are mapped to vectors of real numbers. Conceptually it involves a mathematical embedding from a space with one dimension per word to a continuous vector space with a much lower dimension. Output: Word embeddings are useful for tasks such as sentiment analysis, text classification, predicting the next word in a sequence, and understanding synonyms and analogies.
Rating [Retrieval]
Explanation The output discusses the applications of word embeddings, while the evidence only discusses the definitions of word embeddings and how they work. Therefore, we need to retrieve other evidence to verify whether the output is correct or not.

Table 9: Instructions and demonstrations for Retrieve aspect given the input, preceding generations, and retrieved passages.

Instructions
You'll be provided with an instruction, along with evidence and possibly some preceding sentences. When there are preceding sentences, your focus should be on the sentence that comes after them. Your job is to determine if the evidence is relevant to the initial instruction and the preceding context, and provides useful information to complete the task described in the instruction. If the evidence meets this requirement, respond with [Relevant]; otherwise, generate [Irrelevant].

Instruction Given four answer options, A, B, C, and D, choose the best answer.
Input Earth's rotating causes
A: the cycling of AM and PM
B: the creation of volcanic eruptions
C: the cycling of the tides
D: the creation of gravity
Evidence Rotation causes the day-night cycle which also creates a corresponding cycle of temperature and humidity creates a corresponding cycle of temperature and humidity. Sea level rises and falls twice a day as the earth rotates.
Rating [Relevant]
Explanation The evidence explicitly mentions that the rotation causes a day-night cycle, as described in the answer option A.

Instruction age to run for US House of Representatives
Evidence The Constitution sets three qualifications for service in the U.S. Senate: age (at least thirty years of age); U.S. citizenship (at least nine years); and residency in the state a senator represents at the time of election.
Rating [Irrelevant]
Explanation The evidence only discusses the ages to run for the US Senate, not for the House of Representatives.

Table 10: Instructions and demonstrations for ISREL aspect given the input only.

Instructions
You will receive an instruction, evidence, and output, and optional preceding sentences. If the preceding sentence is given, the output should be the sentence that follows those preceding sentences. Your task is to evaluate if the output is fully supported by the information provided in the evidence.
Use the following entailment scale to generate a score:
- [Fully supported] - All information in output is supported by the evidence, or extractions from the evidence. This is only applicable when the output and part of the evidence are almost identical.
- [Partially supported] - The output is supported by the evidence to some extent, but there is major information in the output that is not discussed in the evidence. For example, if an instruction asks about two concepts and the evidence only discusses either of them, it should be considered a [Partially supported].
- [No support / Contradictory] - The output completely ignores evidence, is unrelated to the evidence, or contradicts the evidence. This can also happen if the evidence is irrelevant to the instruction.

Make sure to not use any external information/knowledge to judge whether the output is true or not. Only check whether the output is supported by the evidence, and not whether the output follows the instructions or not.

Instruction Explain the use of word embeddings in Natural Language Processing.
Preceding sentences Word embeddings are one of the most powerful tools available for Natural Language Processing (NLP). They are mathematical representations of words or phrases in a vector space, allowing similarities between words and the context in which they are used to be measured.
Output Word embeddings are useful for tasks such as sentiment analysis, text classification, predicting the next word in a sequence, and understanding synonyms and analogies.
Evidence Word embedding
Word embedding is the collective name for a set of language modeling and feature learning techniques in natural language processing (NLP) where words or phrases from the vocabulary are mapped to vectors of real numbers. Conceptually it involves a mathematical embedding from a space with one dimension per word to a continuous vector space with a much lower dimension. Methods to generate this mapping include neural networks, dimensionality reduction on the word co-occurrence matrix, probabilistic models, explainable knowledge base method, and explicit representation in terms of the context in which words appear. Word and phrase embeddings, when used as the underlying input representation, have been shown to boost the performance in NLP tasks such as syntactic parsing, sentiment analysis, next token predictions as well and analogy detection.
Score [Fully supported]
Explanation The output sentence discusses the application of word embeddings, and the evidence mentions all of the applications syntactic parsing, sentiment analysis, next token predictions as well as analogy detection as the applications. Therefore, the score should be [Fully supported].

Table 11: Instructions and demonstrations for IsSup tokens.

Instructions
Given an instruction and an output, rate whether the response appears to be a helpful and informative answer to the query, from 1 (lowest) - 5 (highest). We call this score perceived utility. The detailed criterion is as follows: 5: The response provides a complete, highly detailed, and informative response to the query, fully satisfying the information needs. 4: The response mostly fulfills the need in the query, while there can be some minor improvements such as discussing more detailed information, having better structure of the response, or improving coherence. 3: The response is acceptable, but some major additions or improvements are needed to satisfy users' needs. 2: The response still addresses the main request, but it is not complete or not relevant to the query. 1: The response is barely on-topic or completely irrelevant.

Instruction Who is the current prime minister of the UK as of 2023?
Output Boris Johnson was the prime minister of the UK from 2019 - 2022.
Perceived utility 2
Explanation While the output provides a factually correct statement about the UK prime minister from 2019 to 2022, this instruction asks who the prime minister is as of 2023, so it doesn't answer the instruction. Therefore, the utility is 2.

Instruction Given a description of a travel destination, recommend 10 tourist attractions with detailed descriptions of each. The travel destination is Tokyo, Japan.
Output Tokyo is a vibrant city full of exciting tourist attractions. Some must-see sights include the Tokyo Skytree, Tokyo Disneyland, Sensoji Temple, Meiji Shrine, Tsukiji Fish Market, Harajuku, and Shinjuku Gyoen.
Perceived utility 3
Explanation This output doesn't provide descriptions of each attraction and the number of the attractions is also less than 10. While this output partially answers the instructions, it doesn't match the instructions strictly.

Table 12: Instructions and demonstrations for IsUse tokens.