Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection

Máté Gedeon

October 9, 2024

Outline

- Introduction to (Self-)RAG
- Inner workings of Self-RAG
- How did they train it?
- How well does it perform?
- What can we learn from it? (Discussion)

RAG Introduction

Problem: Factual errors in LLM outputs
Solution: Retrieval-Augmented Generation (RAG)

- Augments LLM input with relevant retrieved passages
- Reduces factual errors in knowledge-intensive tasks

Challenges:

- Unnecessary or off-topic passages
  - **Reason:** retrieving passages indiscriminately
- Inconsistent output with retrieved information
  - **Reason:** the models are not explicitly trained to follow facts

RAG Introduction

Problem: Factual errors in LLM outputs
Solution: Retrieval-Augmented Generation (RAG)

- Augments LLM input with relevant retrieved passages
- Reduces factual errors in knowledge-intensive tasks

Challenges:

- Unnecessary or off-topic passages
  - **Reason:** retrieving passages indiscriminately
- Inconsistent output with retrieved information
  - **Reason:** the models are not explicitly trained to follow facts

Self-RAG Introduction

Goal: Improve LLM factual accuracy without sacrificing versatility

Method: On-demand retrieval + Self-reflection

- Generates task output + reflection tokens
- Reflection tokens: indicate *need for retrieval* or *critique* quality

Motivation: Inspired by reinforcement learning (RLHF)

Self-RAG Introduction

Goal: Improve LLM factual accuracy without sacrificing versatility
Method: On-demand retrieval + Self-reflection

- Generates task output + reflection tokens
- Reflection tokens: indicate *need for retrieval* or *critique* quality

Motivation: Inspired by reinforcement learning (RLHF)

Self-RAG Introduction

Goal: Improve LLM factual accuracy without sacrificing versatility
Method: On-demand retrieval + Self-reflection

- Generates task output + reflection tokens
- Reflection tokens: indicate *need for retrieval* or *critique* quality

Motivation: Inspired by reinforcement learning (RLHF)

Self-RAG Workflow

Three phases:

- **Retrieval Phase:** Determines if retrieval is needed
  - If *yes*, it outputs a retrieval token that calls a retriever model on demand
- **Generation Phase:** Uses relevant passages to generate output
- **Critic Phase:** Critiques output and chooses the best one

Customization:

- High factuality tasks: frequent retrieval
- Open-ended tasks: less retrieval, prioritize creativity

Self-RAG Workflow

Three phases:

- **Retrieval Phase:** Determines if retrieval is needed
    - If *yes*, it outputs a retrieval token that calls a retriever model on demand
- **Generation Phase:** Uses relevant passages to generate output
- **Critic Phase:** Critiques output and chooses the best one

Customization:

- High factuality tasks: frequent retrieval
- Open-ended tasks: less retrieval, prioritize creativity

Self-RAG Workflow

Three phases:

- **Retrieval Phase:** Determines if retrieval is needed
  - If *yes*, it outputs a retrieval token that calls a retriever model on demand
- **Generation Phase:** Uses relevant passages to generate output
- **Critic Phase:** Critiques output and chooses the best one

Customization:

- High factuality tasks: frequent retrieval
- Open-ended tasks: less retrieval, prioritize creativity

Self-RAG Workflow

Three phases:

- **Retrieval Phase:** Determines if retrieval is needed
  - If *yes*, it outputs a retrieval token that calls a retriever model on demand
- **Generation Phase:** Uses relevant passages to generate output
- **Critic Phase:** Critiques output and chooses the best one

Customization:

- High factuality tasks: frequent retrieval
- Open-ended tasks: less retrieval, prioritize creativity

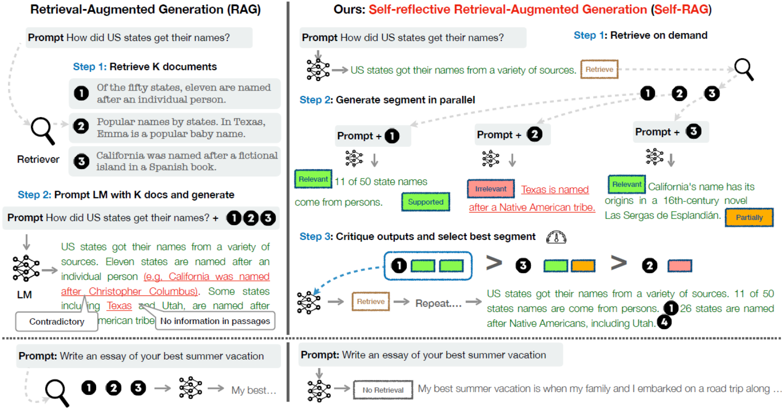


Figure: Traditional RAG vs. Self-RAG

How is this possible?

Self-RAG trains an arbitrary LM to generate text with reflection tokens (next token prediction)

How is this possible?

Self-RAG trains an arbitrary LM to generate text with reflection tokens (next token prediction)

Key Components:

- **Reflection tokens:** Critically assess generation quality
- **Training:** LM trained with interleaved reflection tokens and retrieved passages

How is this possible?

Self-RAG trains an arbitrary LM to generate text with reflection tokens (next token prediction)

Key Components:

- **Reflection tokens:** Critically assess generation quality
- **Training:** LM trained with interleaved reflection tokens and retrieved passages

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.

Critic Model:

- Inserts reflection tokens offline, reducing overhead
- Trained on input-output pairs and reflection tokens (collected by LM)

(More) Formally

- Let M be an arbitrary LM

(More) Formally

- Let M be an arbitrary LM
- Input x, we train M to sequentially generate textual outputs

(More) Formally

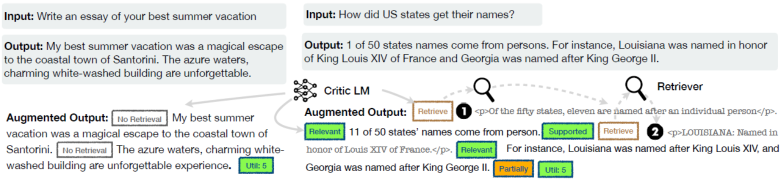
- Let M be an arbitrary LM
- Input x, we train M to sequentially generate textual outputs
- output y is sequentially generated consisting of multiple segments $y = [y_1, \ldots, y_T]$, where y_t indicates a sequence of tokens for the t-th segment

(More) Formally

- Let M be an arbitrary LM
- Input x, we train M to sequentially generate textual outputs
- output y is sequentially generated consisting of multiple segments $y = [y_1, \ldots, y_T]$, where y_t indicates a sequence of tokens for the t-th segment
- Generated tokens in y_t include text from the original vocabulary as well as the reflection tokens.

(More) Formally

- Let M be an arbitrary LM
- Input x, we train M to sequentially generate textual outputs
- output y is sequentially generated consisting of multiple segments $y = [y_1, \ldots, y_T]$, where y_t indicates a sequence of tokens for the t-th segment
- Generated tokens in y_t include text from the original vocabulary as well as the reflection tokens.
- Two models: *critic* model and *generator* model



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 $\boxed{\text{Retrieve}}$ given $(x, y_{<t})$

if $\boxed{\text{Retrieve}}$ == `Yes` **then**

    Retrieve relevant text passages $\mathbf{D}$ using $\mathcal{R}$ given (x, y_{t-1}) ▷ Retrieve

    $\mathcal{M}$ predicts $\boxed{\text{IsRel}}$ given x, d and y_t given $x, d, y_{<t}$ for each $d \in \mathbf{D}$ ▷ Generate

    $\mathcal{M}$ predicts $\boxed{\text{IsSup}}$ and $\boxed{\text{IsUse}}$ given x, y_t, d for each $d \in \mathbf{D}$ ▷ Critique

    Rank y_t based on $\boxed{\text{IsRel}}$, $\boxed{\text{IsSup}}$, $\boxed{\text{IsUse}}$ ▷ Detailed in Section 3.3

else if $\boxed{\text{Retrieve}}$ == `No` **then**

    $\mathcal{M}_{gen}$ predicts y_t given x ▷ Generate

    $\mathcal{M}_{gen}$ predicts $\boxed{\text{IsUse}}$ given x, y_t ▷ Critique

Figure: Pseudo Code of the Generator Model

Training Process

Two models, the *critic model* and the *generator model*

Critic Model Training:

- Data collection
  - by hand would be expensive → utilizing SOTA LLMs
  - GPT-4 is the best, but API call costs add up quickly, and diminish reproducibility
  - **Solution:** supervised data by prompting GPT-4 to generate reflection tokens and then distill their knowledge into an in-house critic model
  - For different reflection token groups different instruction prompts are used
- GPT-4 prompt example
  - "Given an instruction, make a judgment on whether finding some external documents from the web helps to generate a better response."
  - Is this in agreement with human judgement?
- Full dataset size: 4k-20k supervised training data for each type

Training Process

Two models, the *critic model* and the *generator model*

Critic Model Training:

- Data collection
  - by hand would be expensive → utilizing SOTA LLMs
  - GPT-4 is the best, but API call costs add up quickly, and diminish reproducibility
  - **Solution:** supervised data by prompting GPT-4 to generate reflection tokens and then distill their knowledge into an in-house critic model
  - For different reflection token groups different instruction prompts are used
- GPT-4 prompt example
  - "Given an instruction, make a judgment on whether finding some external documents from the web helps to generate a better response."
  - Is this in agreement with human judgement?
- Full dataset size: 4k-20k supervised training data for each type

Training Process

Two models, the *critic model* and the *generator model*

Critic Model Training:

- Data collection
  - by hand would be expensive → utilizing SOTA LLMs
  - GPT-4 is the best, but API call costs add up quickly, and diminish reproducibility
  - **Solution:** supervised data by prompting GPT-4 to generate reflection tokens and then distill their knowledge into an in-house critic model
  - For different reflection token groups different instruction prompts are used
- GPT-4 prompt example
  - "Given an instruction, make a judgment on whether finding some external documents from the web helps to generate a better response."
  - Is this in agreement with human judgement?
- Full dataset size: 4k-20k supervised training data for each type

Training Process

Two models, the *critic model* and the *generator model*

Critic Model Training:

- Data collection
  - by hand would be expensive → utilizing SOTA LLMs
  - GPT-4 is the best, but API call costs add up quickly, and diminish reproducibility
  - **Solution:** supervised data by prompting GPT-4 to generate reflection tokens and then distill their knowledge into an in-house critic model
  - For different reflection token groups different instruction prompts are used
- GPT-4 prompt example
  - "Given an instruction, make a judgment on whether finding some external documents from the web helps to generate a better response."
  - Is this in agreement with human judgement?
- Full dataset size: 4k-20k supervised training data for each type

Training Process

Two models, the *critic model* and the *generator model*

Critic Model Training:

- Data collection
  - by hand would be expensive → utilizing SOTA LLMs
  - GPT-4 is the best, but API call costs add up quickly, and diminish reproducibility
  - **Solution:** supervised data by prompting GPT-4 to generate reflection tokens and then distill their knowledge into an in-house critic model
  - For different reflection token groups different instruction prompts are used
- GPT-4 prompt example
  - "Given an instruction, make a judgment on whether finding some external documents from the web helps to generate a better response."
  - Is this in agreement with human judgement?
- Full dataset size: 4k-20k supervised training data for each type

Training Process

Two models, the *critic model* and the *generator model*

Critic Model Training:

- Data collection
  - by hand would be expensive → utilizing SOTA LLMs
  - GPT-4 is the best, but API call costs add up quickly, and diminish reproducibility
  - **Solution:** supervised data by prompting GPT-4 to generate reflection tokens and then distill their knowledge into an in-house critic model
  - For different reflection token groups different instruction prompts are used
- GPT-4 prompt example
  - "Given an instruction, make a judgment on whether finding some external documents from the web helps to generate a better response."
  - Is this in agreement with human judgement?
- Full dataset size: 4k-20k supervised training data for each type

Training Process

Two models, the *critic model* and the *generator model*

Critic Model Training:

- Data collection
  - by hand would be expensive $\rightarrow$ utilizing SOTA LLMs
  - GPT-4 is the best, but API call costs add up quickly, and diminish reproducibility
  - **Solution:** supervised data by prompting GPT-4 to generate reflection tokens and then distill their knowledge into an in-house critic model
  - For different reflection token groups different instruction prompts are used
- GPT-4 prompt example
  - "Given an instruction, make a judgment on whether finding some external documents from the web helps to generate a better response."
  - Is this in agreement with human judgement?
- Full dataset size: 4k-20k supervised training data for each type

Training Setup

Training

- initialize C with a pre-trained LM and train it on collected data
- Llama 2-7B is used for C initialization
- higher than 90% agreement with GPT-4-based predictions (on most reflection token categories)

Computational resources

- 4 Nvidia A100 with 80GB memory for training
- maximum token length is set to be 2,048 for 7B model, 1524 for 13B model
- Deepspeed stage 3 to conduct multi-GPU distributed training
- FlashAttention is used to make the long-context training more efficient
- Inference of the trained models is ran using 1-2 Quadro RTX 6000 GPUs with 24GB memory

Training Setup

Training

- initialize C with a pre-trained LM and train it on collected data
- Llama 2-7B is used for C initialization
- higher than 90% agreement with GPT-4-based predictions (on most reflection token categories)

Computational resources

- 4 Nvidia A100 with 80GB memory for training
- maximum token length is set to be 2,048 for 7B model, 1524 for 13B model
- Deepspeed stage 3 to conduct multi-GPU distributed training
- FlashAttention is used to make the long-context training more efficient
- Inference of the trained models is ran using 1-2 Quadro RTX 6000 GPUs with 24GB memory

Evaluation

Metrics:

- Correctness, factuality, fluency

Tasks:

- **Closed-set:** fact verification dataset about public health (PubHealth), multiple-choice reasoning dataset created from scientific exams (ARC-Challenge)
- **Open-domain QA:** open-domain question answering (PopQA, TriviaQA)
- **Long-form:** biography generation task, long-form QA task (ALCE-ASQA)
  - *used metric:* FactScore to evaluate biographies, metrics of correctness (str-em), fluency based on MAUVE, and citation precision and recall for ASQA.

Evaluation

Metrics:

- Correctness, factuality, fluency

Tasks:

- **Closed-set:** fact verification dataset about public health (PubHealth), multiple-choice reasoning dataset created from scientific exams (ARC-Challenge)
- **Open-domain QA:** open-domain question answering (PopQA, TriviaQA)
- **Long-form:** biography generation task, long-form QA task (ALCE-ASQA)
  - *used metric:* FactScore to evaluate biographies, metrics of correctness (str-em), fluency based on MAUVE, and citation precision and recall for ASQA.

Baseline Models

Without retrieval

- publicly available LLMs (Llama2 7B,13B)
- instruction-tuned models (Alpaca 7B,13B)
- models trained and reinforced using private data (ChatGPT, Llama2-chat13B)

  Concurrent model: CoVE65B, which introduces iterative prompt engineering to improve the factuality of LLM generations

With retrievals

- standard RAG baselines: an LM (Llama2, Alpaca) generates output given the query prepended with the top retrieved documents using the same retriever as in our system
- *Llama2-FT*, where Llama2 is fine-tuned on all training data used for Self-RAG without the reflection tokens or retrieved passages
- Retrieval-augmented baselines with LMs trained with private data: Ret-ChatGPT, Ret-Llama2-chat, perplexity.ai

Baseline Models

Without retrieval

- publicly available LLMs (Llama2 7B,13B)
- instruction-tuned models (Alpaca 7B,13B)
- models trained and reinforced using private data (ChatGPT, Llama2-chat13B)

  Concurrent model: CoVE65B, which introduces iterative prompt engineering to improve the factuality of LLM generations

With retrievals

- standard RAG baselines: an LM (Llama2, Alpaca) generates output given the query prepended with the top retrieved documents using the same retriever as in our system
- *Llama2-FT*, where Llama2 is fine-tuned on all training data used for Self-RAG without the reflection tokens or retrieved passages
- Retrieval-augmented baselines with LMs trained with private data: Ret-ChatGPT, Ret-Llama2-chat, perplexity.ai

Evaluation

LM	Short-form PopQA (acc)	TQA (acc)	Closed-set Pub (acc)	ARC (acc)	Long-form generations (with citations) Bio (FS)	ASQA (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**

Figure: Evaluation Metrics

Results

- Self-RAG outperforms retrieval-augmented ChatGPT on four tasks, Llama2-chat and Alpaca on all tasks.

Without Retrieval:

- SELF-RAG (bottom two rows) shows a substantial performance advantage over supervised fine-tuned LLMs on all tasks.
- Outperforms ChatGPT in PubHealth, PopQA, biography generation, and ASQA (Rouge and MAUVE)
- Outperforms concurrent CoVE (Dhuliawala et al., 2023) on the bio generation task with 7B and 13B models

Results

- Self-RAG outperforms retrieval-augmented ChatGPT on four tasks, Llama2-chat and Alpaca on all tasks.

Without Retrieval:

- SELF-RAG (bottom two rows) shows a substantial performance advantage over supervised fine-tuned LLMs on all tasks.
- Outperforms ChatGPT in PubHealth, PopQA, biography generation, and ASQA (Rouge and MAUVE)
- Outperforms concurrent CoVE (Dhuliawala et al., 2023) on the bio generation task with 7B and 13B models

Results

With Retrieval:

- SELF-RAG outperforms existing RAG, obtaining the best performance among non-proprietary LM-based models.
- Powerful retrieval-augmented LMs like Llama2-chat and Alpaca show significant gains but fail to improve citation accuracy or performance on tasks like PubHealth and ARC-Challenge.
- SELF-RAG shows higher citation precision and recall than all models except ChatGPT, bridging the performance gap.
- Llama2-FT7B lags behind SELF-RAG, suggesting gains are not solely from training data but the framework itself.

Discussion

- What can we learn from this?
- Can we use any of it?
- All models and training code is available