

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

CMPE 252 - Section 03

Ayushi Bhatnagar, Maxim Dokukin, Krushna Thakkar(015262507)

Content

1 Paper: Intro



2 Paper: Problem Definition



3 Paper: Input -> Output



4 Paper: Methodology



5 Paper: Results



6 Our Contribution



Introduction

“SELF-RAG bridges generation and reasoning — making LLMs their own fact-checkers.”

Overview

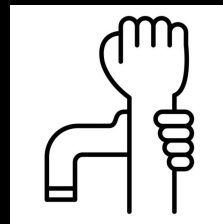
- Large Language Models (LLMs) excel at generating fluent text but often hallucinate facts or misuse retrieved information.
- Retrieval-Augmented Generation (RAG) enhances factual grounding by adding external documents, yet it still lacks *self-awareness* about when and what to retrieve.

Paper Theory

SELF-RAG introduces a self-reflective mechanism, enabling the model to:

- Decide when and what to retrieve information dynamically.
- Critique retrieved passages for relevance, support, and usefulness.
- Control generation through internal reflection tokens (ISREL, ISSUP, ISUSE).

This leads to more accurate, verifiable, and efficient knowledge-grounded generation.



Self-RAG Architecture

Retrieval-Augmented Generation (RAG)

Prompt How did US states get their names?

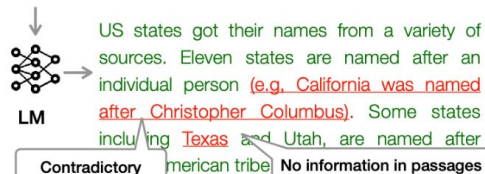
Step 1: Retrieve K documents

- 1 Of the fifty states, eleven are named after an individual person.
- 2 Popular names by states. In Texas, Emma is a popular baby name.
- 3 California was named after a fictional island in a Spanish book.

Retriever

Step 2: Prompt LM with K docs and generate

Prompt How did US states get their names? + 1 2 3



Prompt: Write an essay of your best summer vacation



Ours: Self-reflective Retrieval-Augmented Generation (Self-RAG)

Prompt How did US states get their names?

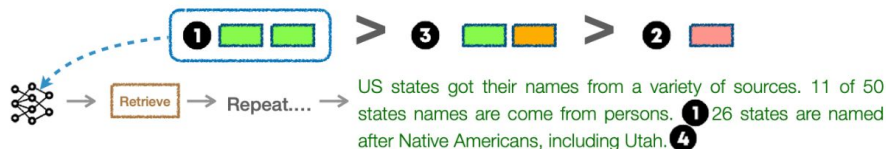
Step 1: Retrieve on demand



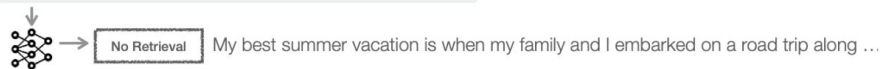
Step 2: Generate segment in parallel



Step 3: Critique outputs and select best segment



Prompt: Write an essay of your best summer vacation



Problem Definition

Objective:

- Train a large language model that can dynamically alternate between generation and retrieval during inference, guided by internal self-reflection

Challenge addressed:

- When to retrieve (avoid unnecessary retrieval)
- How to integrate retrieved documents effectively
- How to perform self-evaluation and correction in-context

The model must learn to:

- Detect uncertainty or factual gaps
 - Issue retrieve tokens (e.g. <RET>)
 - Integrate new context (retrieved passages)
 - Generate a final answer that it can later critique using reflection tokens (e.g. <CRITIQUE> , <REFLECT>)
-

Problem Input & Output

Input:

- A user query or prompt (e.g. "When did Apollo 11 land on the Moon")
- A retrieval corpus (Wikipedia, document database, etc.)
- Optionally retrieved text chunks if the model decides to retrieve

Output:

- A generated text sequences that includes
 - The final answer to the query
 - Optional reflection tokens where the model critiques itself
 - Metadata about retrieval usage (e.g. retrieved docs)

Example:

Query: Who discovered penicillin?

Model output: <RET> [Retrieved: Alexander Fleming, 1928]

Penicillin was discovered by Alexander Fleming in 1928.

<REFLECT> This answer is well-supported by the retrieved evidence.

Methodology, Components & Training

Goal: one LM that decides when to retrieve, generates, and self-critiques.

Reflection tokens

- Retrieve: yes, no, continue
- ISREL: relevant, irrelevant
- ISSUP: fully, partially, no support
- ISUSE: 1 to 5 utility

Models

- C: critic to predict reflection tokens
- M: generator to produce text plus tokens
- R: retriever over Wikipedia and web

Training data creation

- Prompt GPT-4 to label reflection tokens on input-output pairs
 - Fine-tune C on these labels
 - Use C to insert tokens and attach top K retrieved passages per sentence
 - Mask passage spans from loss
 - Train M with next-token objective over text plus tokens
-

Methodology, Inference & Control

Per segment loop

- M predicts Retrieve given input and prior text
- If yes: R fetches K passages
- For each passage in parallel: M generates a candidate segment and predicts ISREL, ISSUP, ISUSE

Selection

- Score candidate = log prob of segment plus weighted scores of ISREL, ISSUP, ISUSE
- Segment-level beam search chooses the next segment

Controls

- Retrieval threshold on Retrieve=yes probability
- Weights trade off support vs utility and fluency
- Hard constraint option: drop candidates with ISSUP=no support
- Continue reuses the same evidence across segments

Outputs

Segment-level citations with self-assessed support and relevance

Results

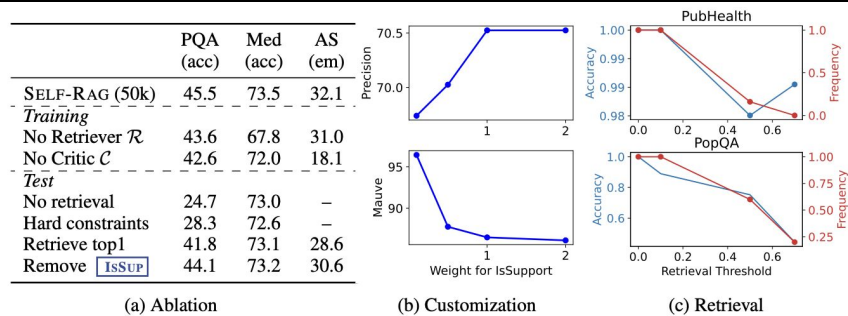


Figure 3: **Analysis on SELF-RAG:** (a) **Ablation studies** for key components of SELF-RAG training and inference based on our 7B model. (b) **Effects of soft weights** on ASQA citation precision and Maue (fluency). (c) **Retrieval frequency** and *normalized* accuracy on PubHealth and PopQA.

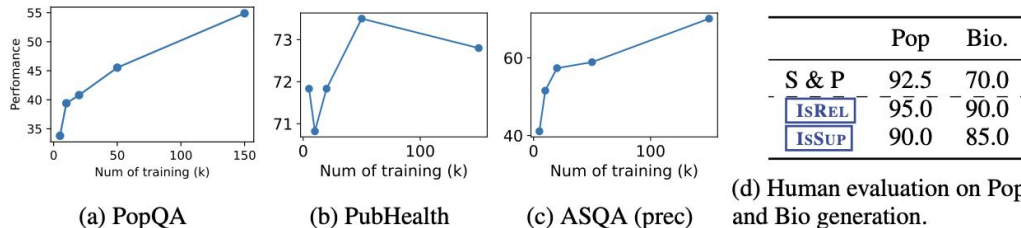


Figure 4: **Training scale and Human analysis:** (a) (b) (c) **Training scale analysis** shows the effect of the training data scale on PopQA, PubHealth and ASQA (citation precision), respectively. (d) **Human analysis** on SELF-RAG outputs as well as reflection tokens.

Contribution: Prompt Engineering — Why It Matters

- We extend the *SELF-RAG* framework by introducing **Prompt Engineering** as a guiding layer.
- The system prompt steers the model toward **higher factual accuracy, coherence, and natural reasoning style**.
- Addresses a key limitation of vanilla RAG models: *generic or fragmented responses* due to poor prompt structure.
- Moves the model closer to human-like “self-reflective” behavior through **structured instructions** and **clarity-focused templates**

Query: Explain why the sky appears blue using physics concepts.

BASELINE: Original SELF-RAG (no prompt-engineering)

loading requests: 100% 1/1 [00:00<00:00, 126.12it/s]

```
Processed prompts: 100% 1/1 [00:02<00:00, 2.43s/it, est. speed input: 4.94 toks/s, output
```

ding requests: 100% 5/5 [00:00<00:00, 480.05it/s]

Processed prompts: 100% 5/5 [00:04<00:00, 2.02s/it, est. speed input: 63.53 toks/s, output

```
re[Retrieval]<paragraph>[Irrelevant]The sky appears blue because of the way the molecules of the air scatter sunlight.[Continue to Use Evidence]When white light from the sun passes through the ai
```

ENHANCED: Prompt-Engineered + Self-Reflective SELF-RAG

STEP 1 – Initial Generation

ding requests: 100% 1/1 [00:00<00:00, 112.60it/s]

```
Processed prompts: 100% 1/1 [00:04<00:00, 4.13s/it, est. speed input: 34.86 toks/s, output: 1.00 toks/s]
```

Prompt Mode: explanatory

Query: Explain why the sky appears blue using physics concepts.

Generated Output:

Contribution: How (Method & Implementation)

Introduced **custom templates** (qa, explanatory, chain_of_thought, compare_contrast) to frame model intent.

Integrated prompt selection dynamically into the pipeline via:

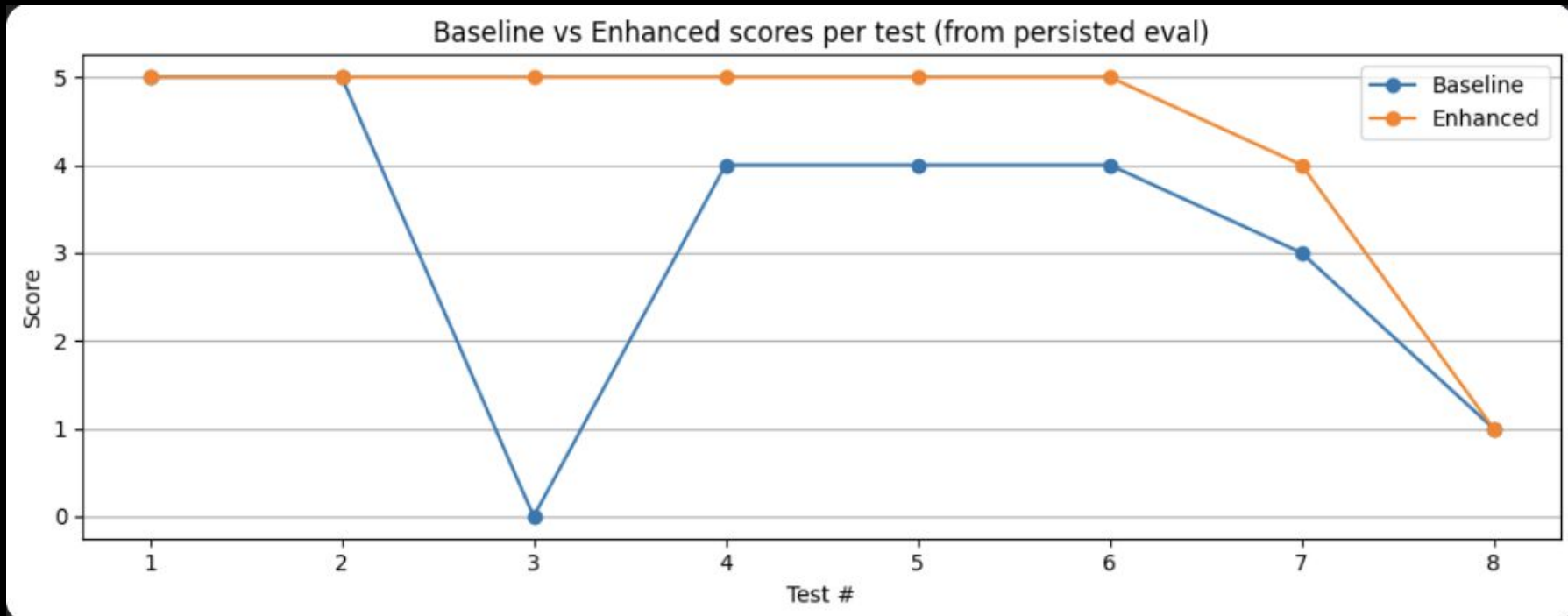
1. `build_prompt()` → constructs contextual prompt.
2. `ask_with_prompt()` → retrieves relevant passages and formats responses.
3. `ask_with_reflection()` → adds *self-critique and refinement*.
4. `ask_full_selfrag()` → performs *evaluation and selection* of best output.

Ensured **retrieval grounding** by embedding top Wikipedia passages directly into prompts.

Added **reflection instructions**: “analyze your answer for accuracy, completeness, and reasoning errors.”

Framework built and tested in **Google Colab with vLLM**, using **SELF-RAG LLaMA 2-7B** on A100 GPU.

Results



Thank you
