

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

Sangho Daniel Kim

www.linkedin.com/in/danieliscoding

Natural Language Processing

2024/05/21

Contents

- Abstract
- Introduction
- Related Work
- Self-RAG
- Experiments
- Results and Analysis
- Conclusion

Abstract

Limitations

기존 LLM들의 한계

오로지 학습한 Parametric Knowledge에만 기반하여 부정확한 답변을 함

"large language models (LLMs) often produce responses containing factual inaccuracies due to their sole reliance on the parametric knowledge they encapsulate."

이에 대한 해결책으로 제시된 RAG(Retrieval-Augmented Generation)

→ 검색된 관련 정보를 LM(언어모델)에게 주는 방식 - 전문성 및 정확도 확보

"...augments LMs with retrieval of relevant knowledge, decreases such issues"

Abstract

Limitations

RAG(Retrieval-Augmented Generation)의 한계점

무조건적으로 검색하고 고정된 수의 검색 정보를 결합하는 행위

"*indiscriminately retrieving and incorporating a fixed number of retrieved passages,*"

Q1. 검색이 꼭 필요한가?

"*regardless of whether retrieval is necessary,*"

Q2. 발췌된 검색 정보의 연관성이 실제로 높은가?

"*or passages are relevant*"

→ LM의 Versatility(다능함) 저하, 품질이 낮은 답변 생성

Abstract

Overview

SELF – RAG (Self-Reflective Retrieval-Augmented Generation)

“Reflective Token”의 도입으로 LM에게 “선택권” 부여

“Generating reflection tokens makes the LM controllable during the inference phase, enabling it to tailor its behavior to diverse task requirements.”

Q1. 검색이 꼭 필요한가?

“regardless of whether retrieval is necessary,”

Q2. 발췌된 문헌 정보의 연관성이 실제로 높은가?

“or passages are relevant”

→ A1. 검색 실시 여부를 선택 사항으로 변경

*“Adaptively retrieves passages **on-demand**,”*

→ A2. 검색된 부분과 생성된 부분에 대한 자기 반성 과정

*“...and generates and **reflects** on retrieved passages and its own generation”*

→ 기존 RAG의 한계점 극복 (무조건성, 과도한 의존성)

Introduction

Background

LLM의 변천사

크기를 늘린 모델들의 등장에도 불구하고 Factual Error 문제를 해결하지 못함

...struggle with factual errors (Mallen et al., 2023; Min et al., 2023) despite their increased model and data scale (Ouyang et al., 2022).

→ RAG의 등장으로 '정보 검색'을 통해 Factual Error 문제 부분적으로 해결

...augment the input of LLMs with relevant retrieved passages, reducing factual errors

→ 오히려 LLM의 기동성(다재다능함)을 저해, 관련성이 떨어지는 정보들을 조합

...may hinder the versatility of LLMs or introduce unnecessary or off-topic passages that lead to low-quality generations (Shi et al., 2023)

원인

1. 기반한 정보가 실제로 유관한지에 관계없이 문서를 검색하기 때문

since they retrieve passages indiscriminately regardless of whether the factual grounding is helpful.

2. 검색된 정보와 Output이 일관적인지 확실하지 않음 (문서에서 추출된 정보들을 서로 비교하는 방법을 모름)

The output is not guaranteed to be consistent with retrieved relevant passages since the models are not explicitly trained to leverage and follow facts from provided passages.

(Gao et al., 2023)

Introduction

SELF-RAG

Self-Reflective Retrieval-augmented Generation (SELF-RAG)

Demand Retrieval & Self – Reflection을 통한 Factual Accuracy 증진

its factual accuracy without hurting its versatility, via on-demand retrieval and self-reflection.

답변 생성 과정을 성찰하도록 Task Output 과 간헐적인 Special Token(Reflection Token)을 생성

...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 Token = Retrieval Token + Critique Token

Reflection tokens are categorized into retrieval and critique tokens

→ 정보 검색의 필요성과 답변 생성 퀄리티 확인

...to indicate the need for retrieval and its generation quality respectively

Introduction

SELF-RAG

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.

Step 2: Prompt LM with K docs and generate

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

LM
US states got their names from a variety of sources. Eleven states are named after an individual person (e.g., California was named after Christopher Columbus). Some states including Texas and Utah, are named after American tribe. No information in passages

Prompt: Write an essay of your best summer vacation

1 2 3 → My best...



TRAIN AND TEST

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

Prompt How did US states get their names?

Step 1: Retrieve on demand

Retriever → US states got their names from a variety of sources. Retrieve

1 2 3

Step 2: Generate segment in parallel

Prompt + 1

Relevant
11 of 50 state names come from persons.
Supported

Prompt + 2

Irrelevant
Texas is named after a Native American tribe.

Prompt + 3

Relevant
California's name has its origins in a 16th-century novel Las Sergas de Esplandián.
Partially

Step 3: Critique outputs and select best segment

1 > 3 > 2

Retriever → Repeat... → US states got their names from a variety of sources. 11 of 50 states names are come from persons. 26 states are named after Native Americans, including Utah.

Prompt: Write an essay of your best summer vacation

No Retrieval → My best summer vacation is when my family and I embarked on a road trip along ...

1단계 : 검색을 통한 정보 검색이 도움이 되는지 판별

determines if augmenting the continued generation with retrieved passages would be helpful.

YES : Retrieval Token 생성, Retriever Model 실행

It outputs a retrieval token that calls a retriever model on demand (Step 1).

NO : LLM만을 사용하여 정보 검색 없이 답변 생성

2단계 : 병렬적으로 검색 정보 연관성 계산, 출력값 생성

Subsequently, SELF-RAG concurrently processes multiple retrieved passages, evaluating their relevance and then generating corresponding task outputs (Step 2).

3단계 : Critique Token 생성, 각 Segment 평가, 사실성과 품질 면에서 최종 출력값 선택

It then generates critique tokens to criticize its own output and choose best one (Step 3) in terms of factuality and overall quality.

구분점

- 지속적으로 고정된 수의 문서 검색 (필요성 고려 유무)
- 단계별 결과값 품질 측정의 단계
- 각 정보의 출처를 기재 여부 (사실 확인 용이성)

Introduction

SELF-RAG

비평 모델 Critic Model (*Critique 아님)

반성 토큰(Reflection Token)을 예측하기 위한 모델

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.

모델 구축 : Reflection Token이 삽입된 Text와 Retrieved Passages를 가지고 생성형 언어 모델 학습

We train our generator LM on a diverse collection of text interleaved with reflection tokens and retrieved passages.

Stage 1. GPT-4를 통해 생성된 (Input, Output, Corresponding Reflection Token) 조합으로 판단 모델을 지도 학습

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).

Stage 2. 이 학습된 판단 모델(=비평 모델)을 통해 Generated Segment (Step2)에 대해 자체 평가를 실시(Token 부여) 가능

...our trained LM uses critique tokens to assess its own predictions after each generated segment as an integral part of the generation output.

- 학습된 비평 모델은 전체 아키텍처에 Offline 형태로 포함되어 텍스트에 반성 토큰(Reflection Token) 부여의 역할
 - 간접적인 처리 시간(Overhead) 감소

Introduction

SELF-RAG

반성 토큰 (Reflection Token) - 강화학습이 적용된 Reward Model에 영향 받음

Reflection tokens, inspired by reward models used in reinforcement learning (Ziegler et al., 2019; Ouyang et al., 2022)

반성 토큰을 통한 자체 평가 - 텍스트 생성 시작과 생성 과정을 조절하는 Control Token에서 영향 받음

While we draw inspiration from studies that use control tokens to start and guide text generation (Lu et al., 2022; Keskar et al., 2019),

SELF-RAG 만의 특이점 : 뛰어난 적용성 (Customizability)

1. 다양한 적용을 위해 검색 빈도 조정 가능
2. 모델의 행동을 사용자의 선호도에 맞게 조정 가능
 - Segment 수준의 빔 서치(Segment-level Beam Search)와 반성 토큰 확률의 가중 합(Weighted Linear Sum)

Related Work

RAG

LM의 Input값에 검색된 Text 문장을 결합함으로써 Knowledge-intensive Task들에 대한 성능 향상

...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)

최신 연구 동향

- LM의 Input 앞에 고정된 수의 검색 문장을 선 추가 - (Luo et al., 2023)
- Retriever과 LM을 동시에 Pre-train한 다음, Few-shot Fine Tuning 실시 - (Izacard et al., 2022b)
- 적응형(Adaptive) 검색을 통해 생성 중에 필요한 경우에만 검색을 수행 - (Jiang et al., 2023)
- 명명된 엔티티에 대한 API 호출을 생성하는 LLM을 훈련 - (Schick et al., 2023)

한계점

- Runtime Inefficiency - (Mallen et al., 2023)
- 불필요한 문맥에 대한 강인성 - (Shi et al., 2023)
- 출처 미 제시 - (Liu et al., 2023a; Gao et al., 2023)



→ SELF-RAG : 필요할 때만(On-demand) 검색 수행, 반성 토큰을 사용하여 Controlled Generation 및 생성 품질 개선 / 출처 제공

Related Work

Concurrent RAG Work

학습 방법을 다양화하거나 Prompt 작성 전략을 사용해 다양한 방법으로 RAG 성능 개선 시도가 있음

propose new training or prompting strategies to improve widely-adopted RAG approaches.

최신 연구 동향

- Retriever과 LM을 두 단계에 걸쳐 Fine-tune (**Instruction-tuning**) - (Lin et al., 2023)
- 자연어 추론 모델(Natural Language Inference Model) 이용 - (Yoran et al., 2023)
- 검색된 문장들을 Prompt 입력 전 필터링, 또는 압축하는 **요약 모델** 도입 - (Xu et al., 2023)
- **LATS** : 사전 훈련된 언어모델을 통해 질문 응답 작업에 필요한 정보 **트리** 검색 - (Zhou et al., 2023)

→ **SELF-RAG** : 자기 반성(Self-Reflection)을 통한 가장 고품질의 답변 선택 출력, 추가적인 성능 평가(사실성 등), Segment 문장 병렬 처리

Related Work

Training and generating with critics

최신 연구 동향

- Human Feedback 강화학습(RLHF)을 통한 LLM 훈련 - (*Schulman et al., 2017*)
- 다중 보상 모델을 이용한 섬세한(Fine-Grained) RLHF - (*Wu et al., 2023*)
- General Control Token을 이용하여 LM 응답 생성 조절 - (*Lu et al., 2022; Korbak et al., 2023*)
- 자기 평가로 운용되는 Decoding Framework 제작(추리 분야 한정, 검색 없음) - (*Xie et al., 2023*)
- Task Output, Natural Language Feedback, Refined Task Output 반복적 생성 - (*Dhuliawala et al., 2023; Madaan et al., 2023; Paul et al., 2023*)

→ SELF-RAG : RLHF보다 현저히 줄어든 Training Cost (Offline 비평 모델)

Reflection Token은 출처에 기반한 생성 조절(Controllable Generation) 가능

Reflection Token으로 검색 유무 결정과 생성문에 대한 자기 평가 가능 (RLHF은 Human Preference Alignment 집중)

SELF-RAG

SELF-RAG

LEARNING TO RETRIEVE, GENERATE AND CRITIQUE

Self-Reflective Retrieval-Augmented Generation (SELF-RAG)

LLM의 창의성과 다능함은 유지한 체,

Self-Reflection과 Retrieval을 사용하여 LLM의 성능과 사실성을 향상시킨 프레임워크

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.

Self-Reflection : LLM이 필요 시, 검색된 문장들 기반 Text 생성, Special Token으로 출력값 평가

Our end-to-end training lets an LLM generate text informed by retrieved passages, if needed, and criticize the output by learning to generate special tokens.

Reflection Token의 역할

1. 정보 검색의 필요성 판단

These reflection tokens (Table 1) signal the need for retrieval

2. 출력된 Segment들과 Input 간의 [관련성, 지지도, 완성도] 측정

confirm the output's relevance, support, or completeness.

SELF-RAG

PROBLEM FORMALIZATION AND OVERVIEW

- 입력 x : 모델에 주어지는 초기 입력 (Query Input)
- 출력 y : 모델이 생성하는 텍스트, 여러 세그먼트로 구성
- 세그먼트 y_t : 텍스트 출력 y 의 구성 요소, 각 세그먼트는 시퀀스 ($y = [y_1, \dots, y_T]$)
- 검색 결과 d : 검색기를 통해 검색된 문장 집합 요소

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 .

- 입력(x)과 검색 없이 생성된 문장($y_{<t}$) 기반 검색 유무 판별
- 입력(x)과 검색된 문장(d) 간 관련도
- 입력(x)과 문장(d)를 기반으로 생성된 세그먼트(y)와 검색된 문장(d) 간 지지도
- 세그먼트(y)가 입력(x)에 대한 대답으로 적절한지에 대한 정도

Algorithm 1 SELF-RAG Inference

Require: Generator LM \mathcal{M} , Retriever \mathcal{R} , Large-scale passage collections $\{d_1, \dots, d_N\}$

- 1: **Input:** input prompt x and preceding generation $y_{<t}$, **Output:** next output segment y_t
- 2: \mathcal{M} predicts **Retrieve** given $(x, y_{<t})$
- 3: **if** **Retrieve** == Yes **then**
 - 4: Retrieve relevant text passages D using \mathcal{R} given (x, y_{t-1}) ▷ Retrieve
 - 5: \mathcal{M} predicts **ISREL** given x, d and y_t given $x, d, y_{<t}$ for each $d \in D$ ▷ Generate
 - 6: \mathcal{M} predicts **ISSUP** and **ISUSE** given x, y_t, d for each $d \in D$ ▷ Critique
 - 7: Rank y_t based on **ISREL**, **ISSUP**, **ISUSE** ▷ Detailed in Section 3.3
- 8: **else if** **Retrieve** == No **then**
 - 9: \mathcal{M}_{gen} predicts y_t given x ▷ Generate
 - 10: \mathcal{M}_{gen} predicts **ISUSE** given x, y_t ▷ Critique

2. 입력(x)과 검색 없이 생성된 문장($y_{<t}$) 기반 검색 유무 판별
3. If 검색을 원한다면,
4. 검색기로 텍스트 집합 D 를 검색
5. 쿼리, 추출 부분, 이들에 기반한 세그먼트 y_t 로 관련도 판별
6. 동일 대상에 대한 지지도와 적절성 판별
7. 관련도, 지지도, 적절성에 따라 랭킹(Ranking) 평가
8. Else 만약 검색을 원하지 않는다면?
9. 비평모델 M 은 쿼리에 대해서 검색 없이 생성문 도출
10. 비평모델 M 은 쿼리와 생성문을 기반으로 적절성 평가

진한 부분 : 우리가 원하는 토큰

the bold text indicates the most desirable critique tokens.

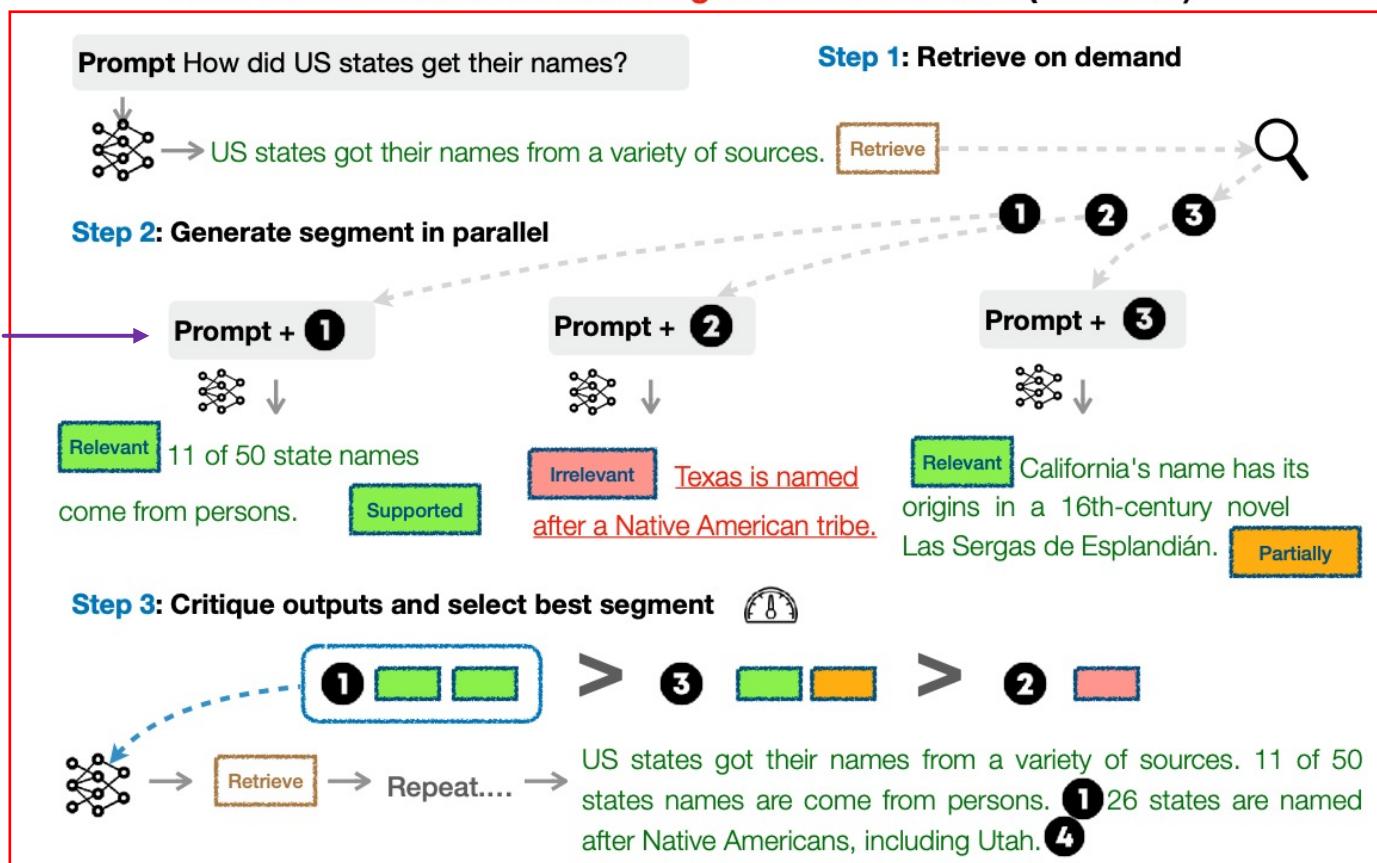
SELF-RAG

PROBLEM FORMALIZATION AND OVERVIEW

- 입력 x : 모델에 주어지는 초기 입력 (Query Input)
- 출력 y : 모델이 생성하는 텍스트, 여러 세그먼트로 구성
- 세그먼트 y_t : 텍스트 출력 y 의 구성 요소, 각 세그먼트는 시퀀스 ($y = [y_1, \dots, y_T]$)
- 검색 결과 d : 검색기를 통해 검색된 문장 집합 요소

Generator Model M

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



- 입력(x)과 검색 없이 생성된 문장($y < t$) 기반 검색 유무 판별
- 입력(x)과 검색된 문장(d) 간 관련도
- 입력(x)과 문장(d)를 기반으로 생성된 세그먼트(y)와 검색된 문장(d) 간 지지도
- 세그먼트(y)가 입력(x)에 대한 대답으로 적절한지에 대한 정도

2. 입력(x)과 검색 없이 생성된 문장($y < t$) 기반 검색 유무 판별
3. If 검색을 원한다면,
4. 검색기로 텍스트 집합 D를 검색
5. 쿼리, 추출 부분, 이들에 기반한 세그먼트 y 로 관련도 판별
6. 동일 대상에 대한 지지도와 적절성 판별
7. 관련도, 지지도, 적절성에 따라 랭킹(Ranking) 평가
8. Else 만약 검색을 원하지 않는다면?
9. 비평모델 M은 쿼리에 대해서 검색 없이 생성문 도출
10. 비평모델 M은 쿼리와 생성문을 기반으로 적절성 평가

SELF-RAG

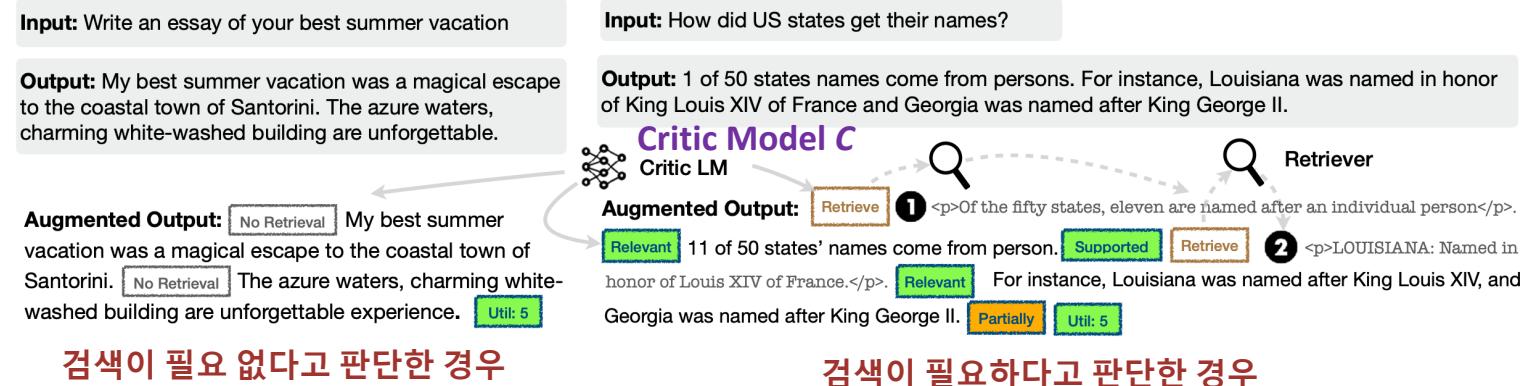
SELF-RAG Training

TRAINING THE CRITIC MODEL

Data Collection for Critic Model C

“Segment에 대한 수동 주석 처리는 너무 비싸다. 우린 가장 좋은 LLM인 GPT-4를 사용한다.”

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 used (Liu et al., 2023b).



SELF-RAG

SELF-RAG Training

TRAINING THE CRITIC MODEL : Predicting the Reflection Token

Data Collection for Critic Model C

GPT-4를 사용해 Reflection Token을 생성하는 지도학습 데이터 확보, In-house 모델 C에 전수(Distill)

We create supervised data by prompting GPT-4 to generate reflection tokens and then distill their knowledge into an in-house C

1. 원래 Training Data에서 Input x 집합과 Output y 집합 중 랜덤으로 샘플 조합 추출 : $\{X^{\text{sample}}, Y^{\text{sample}}\} \sim \{X, Y\}$

For each group of reflection tokens, we randomly sample instances from the original training data.

2. Instruction Prompt로 모델의 역할 설정

As different reflection token groups have their own definitions and input, we use different instruction prompts for them.

3. Few Shot Demonstration I 후, Original Task Input x와 Output y를 가지고 적절한 Reflection Token 예측 : $p(r | I, x, y)$

...followed by few-shot demonstrations | the original task input x and output y to predict an appropriate reflection token as text

Critic Learning

D_critic 훈련 데이터로 (초기화된 LM이 적용된) Model C를 학습, (x, y) 입력 시 Reflection Token r을 예측하도록 훈련

After we collect training data D_{critic} , we initialize C with a pre-trained LM and train it on D_{critic} using a standard conditional language modeling objective, maximizing likelihood.

SELF-RAG

SELF-RAG Training

TRAINING THE GENERATOR MODEL : Generating without M during Inference

Data Collection for Generator

Input, Output 쌍인 (x,y)에서 y를 Retrieval과 Critic Model을 통해 증강하여 지도학습 데이터 생산

Given an input-output pair (x, y) , we augment the original output y using the retrieval and critic models to create supervised data

1. Segment yt 에 대해 Critic Model을 실행하여 Retrieve가 생성에 도움이 되었는지 평가 (Retrieve Token)

For each segment $yt \in y$, we run C to assess whether additional passages could help to enhance generation. If retrieval is required, the retrieval token is added

2. Retrieve를 통해 상위 k개의 문장 집합 D를 만들고 Critic Model을 이용해 관련도 평가 (ISREL Token), 지지도 평가 (ISSUP Token)

C further evaluates whether the passage is relevant. If a passage is relevant, C further evaluates whether the passage supports the model generation

3. 각 세그먼트+Token에 대해 적절성 평가 (ISUSE Token) 후 $(x, y, \text{augmented } y)$ 의 조합을 D_{gen} 에 추가

Critique tokens are appended after the retrieved passage, C predicts the overall utility token, and an augmented output with reflection tokens, original input pair is added to D_{gen}

Generator Learning

D_{gen} 훈련 데이터 샘플로 Model M을 학습, x 입력 시 Reflection Token r과 Output y를 함께 예측하도록 훈련

We train the generator model M by training on the curated corpus augmented with reflection tokens D_{gen} using the standard next token objective

SELF-RAG

SELF-RAG INFERENCE

Adaptive Retrieval and Customization

Reflective Token의 도입으로 Self-Evaluate이 가능해짐 = 다양한 작업 상황에 맞게 행동 조정 가능

*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.*

Adaptive Retrieval with threshold

1. Retrieve 토큰 예측 시, 필요에 따라 동적으로 텍스트 구절을 검색

SELF-RAG dynamically decides when to retrieve text passages by predicting Retrieve Token.

2. 특정 임계값(threshold) 설정, Retrieve=Yes 토큰을 생성할 확률이 이 임계값을 초과하면 검색 실시

*If the probability of generating the Retrieve token **normalized** over all output tokens in Retrieve **surpasses a designated threshold**, we trigger retrieval*

SELF-RAG

SELF-RAG INFERENCE

Adaptive Retrieval and Customization

Reflective Token의 도입으로 Self-Evaluate이 가능해짐 = 다양한 작업 상황에 맞게 행동 조정 가능

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.

Tree-decoding with Critique tokens

1. (세그먼트 단계) 필요에 따라 K 개의 구절을 검색하고, 생성 모델 M 은 각 구절을 병렬로 처리, K 개의 다른 Segment 출력

R retrieves K passages, and the generator M processes each passage in parallel and outputs K different continuation candidates.

2. Segment에 대한 빔 서치(segment-level beam search)를 통해 상위 B 개의 세그먼트 선택, 최종적으로 최상의 시퀀스 반환

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.

3. 이때 각 Segment의 점수는 Critic Score S (Critique Token들의 확률 합)로 저장

critic score S that is the linear weighted sum of the normalized probability of each Critique token type

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

$$\mathcal{S}(\boxed{\text{Critique}}) = \sum_{G \in \mathcal{G}} w^G s_t^G \text{ for } \mathcal{G} = \{\boxed{\text{ISREL}}, \boxed{\text{ISSUP}}, \boxed{\text{ISUSE}}\},$$

행동 조절을 가능하게 하는 Hyperparameter 가장 원하는(긍정) 비평 토큰이 나올 확률

Experiments

Overall

SELF-RAG와 여러 기준 모델을 다양한 작업에서 평가

We conduct evaluations of our SELF-RAG and diverse baselines on a range of downstream tasks,

평가 기준 : 전체적인 정확성, 사실성, 유창성을 측정하는 메트릭 사용

holistically evaluating outputs with metrics designed to assess overall correctness, factuality, and fluency.

모든 실험은 zero-shot 방식으로 수행 (Task 설명을 위한 Few-shot 지시문 포함)

we conduct zero-shot evaluations, where we provide instructions describing tasks without few-shot demonstrations (Wei et al., 2022; Sanh et al., 2022).

Results and Analysis

Overall

1. 검색을 하지 않는 기준 모델과 비교

- 모든 작업에서 유의미한 성능 향상

2. 검색을 하는 기준 모델과 비교

- 전문성을 요구하는 Task에서 특히 우수한 성능

(PubHealth 및 ARC-Challenge)

LM	Short-form		Closed-set		Long-form generations (with citations)					
	PopQA (acc)	TQA (acc)	Pub (acc)	ARC (acc)	Bio (FS)	(em)	(rg)	ASQA (mau)	(pre)	(rec)
<i>LMs with proprietary data</i>										
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	-	-	-	-	-
<i>Baselines without retrieval</i>										
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	-	-	-	-	-
<i>Baselines with retrieval</i>										
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

Results and Analysis

Overall

3. 요소별 분석

- 각 구성 요소가 중요한 역할을 함
- No Retriever 또는 No Critic 모델은 성능이 크게 떨어짐
- SELF-RAG의 세부 조건 기반 우수한 선택 능력 입증

4. 가중치 조정으로 인한 행동 조정

- ISSUP 가중치 증가 시, 기반한 예측률 증가, 유동성 감소
- Threshold 증가 시, Retrieval Frequency, Accuracy 동시 감소

5. 훈련 데이터 크기의 영향

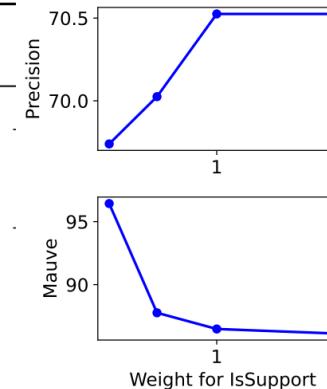
- 더 많은 훈련 데이터는 더 나은 성능으로 이어짐

6. 인간 평가 (Human Evaluation)

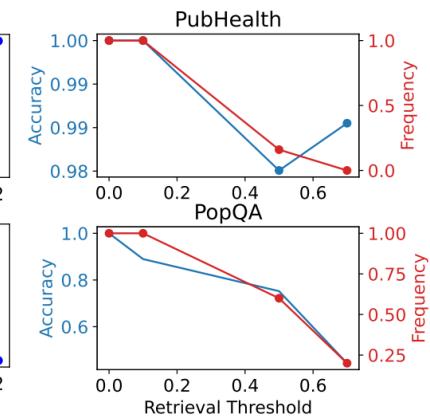
- SELF-RAG 출력물(S&P)과 예측된 Reflection Token의 신뢰성 모두 인간 평가에서 높은 점수 획득

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

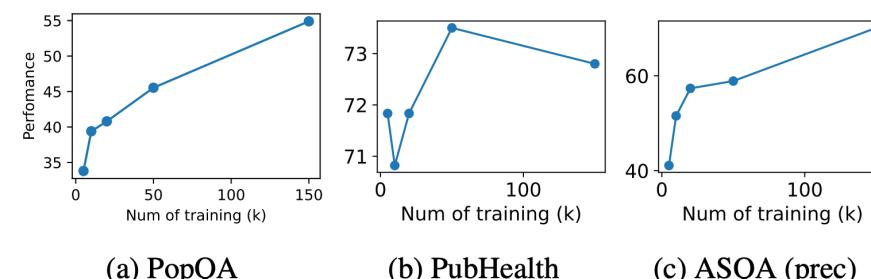
(a) Ablation



(b) Customization



(c) Retrieval



(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.

Conclusion

Overall

Conclusion

SELF-RAG는 Reflection Token을 활용하여 작동 시 LLM의 행동을 사용자 맞춤화 가능

SELF-RAG further enables the tailoring of LM behaviors at test time by leveraging reflection tokens.

**종합 평가 결과, SELF-RAG는 더 많은 파라미터를 가진 LLM이나
기존의 RAG 방식보다 성능이 뛰어나다는 것을 입증함**

*...demonstrate that SELF-RAG significantly outperforms LLMs with more parameters
or with conventional retrieval-augmented generation approaches.*

Ethical Concerns

여전히 Citation들에 불완전하게 기반한 출력이 생성될 수 있음

...it can still generate outputs that are not fully supported by the citations.

수준 높은 Self-Reflection과 세밀하게 조정된 속성으로 극복 가능

...explicit self-reflection and fine-grained attribution may help users verify factual errors in the model outputs.



T R A I N A N D T E S T