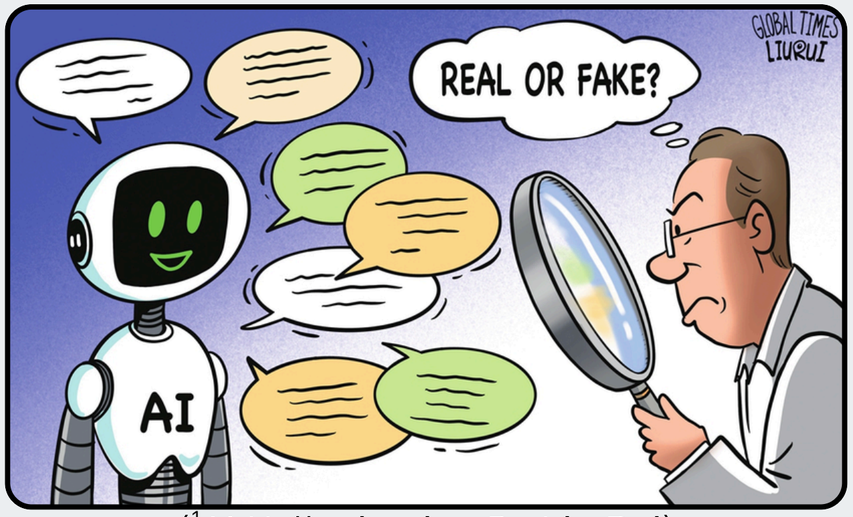
Corrective Retrieval Augmented Generation

Shi-Qi Yan[1*], Jia-Chen Gu[2*], Yun Zhu[3], Zhen-Hua Ling[1]
[1]National Engineering Research Center of Speech and Language Information Processing, University of Science and Technology of China, Hefei, China
[2]Department of Computer Science, University of California, Los Angeles
[3]Google DeepMind

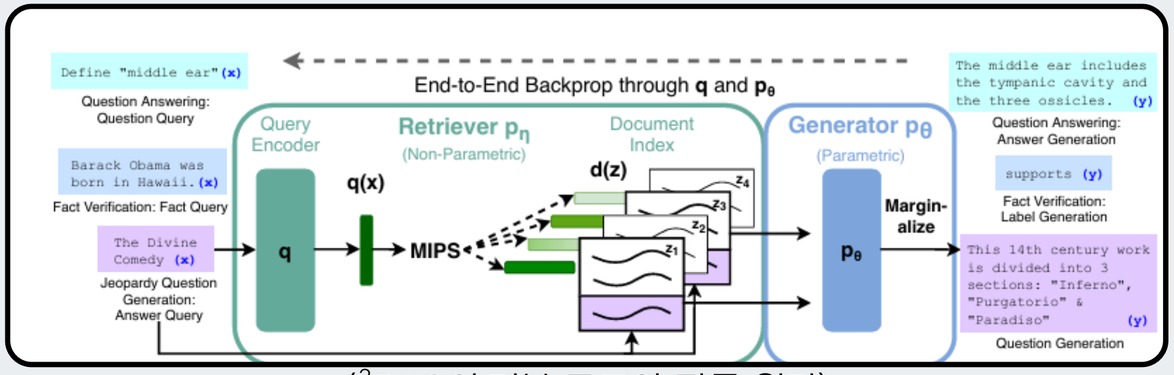
Jan 29, 2024

발제자: 전진구
2025.07.10

1. 연구배경 및 문제정의: LLM의 필연적인 Hallucination과 해결책 RAG



([1]AI Hallucination By Liu Rui)



([2]RAG의 기본 구조와 작동 원리)

대규모 언어 모델(LLM)은 지시를 이해하고 유창한 언어 텍스트를 생성하는 인상적인 능력을 보여주며 점점 더 많은 주목을 받고 있음.

LLM은 **내부에 저장된 지식(매개변수적 지식)**만으로는 생성하는 내용의 사실적 정확성을 100% 보장할 수 없음.

이러한 근본적인 한계 때문에, LLM은 사실과 다른 내용을 그럴듯하게 생성하는 **Hallucination**은 필연적으로 나타남.

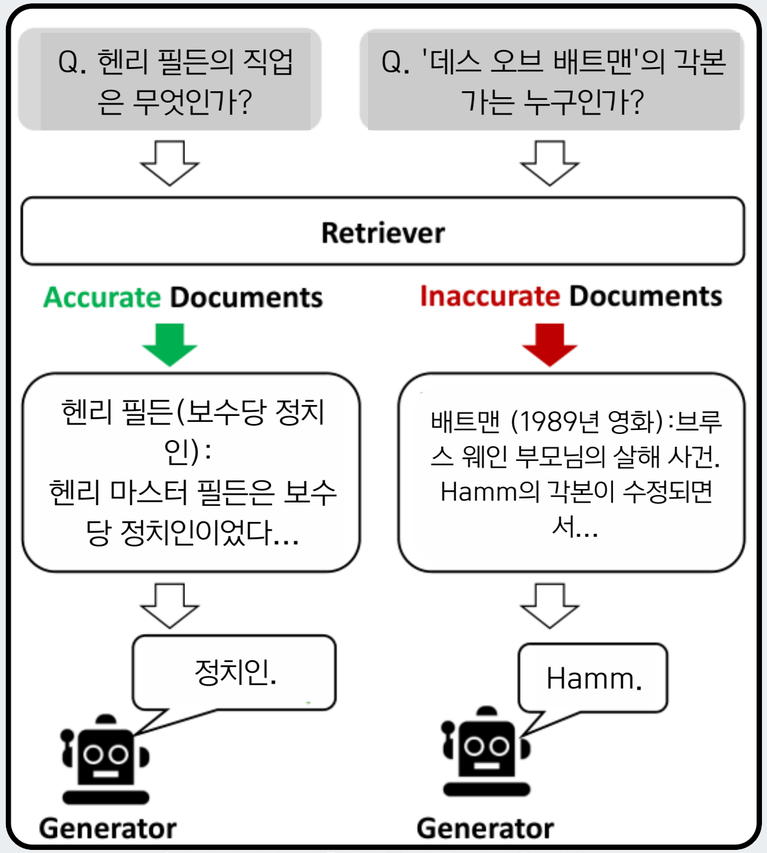
이러한 환각 문제를 해결하기 위한 실용적인 보완책으로
Retrieval-Augmented Generation(RAG) 기술이 제안 됨.

external knowledge corpus(외부 지식 corpus)(예: 위키피디아)에서 질문과 관련된 문서를 **검색(Retrieve)**하고,
이 정보를 LLM의 입력값에 **증강(Augment)** 하여 함께 제공하는 것.

이는 LLM이 자신의 **parametric knowledge** 에만 의존하지 않고,
실시간으로 외부의 최신 정보를 참고하여 답변하게 함으로써
정확도를 높이는 방식.

[1] AI hallucination - Global Times
[2] [2005.11401] Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks

1. 연구배경 및 문제정의: RAG의 치명적 약점



(검색기가 부정확한 문서를 가져올 경우,
틀린 답변을 내놓게 되는 RAG의 예시)

RAG의 효과는 검색된 문서가 얼마나 정확하고 관련성이 높은지에 전적으로 **의존**함.

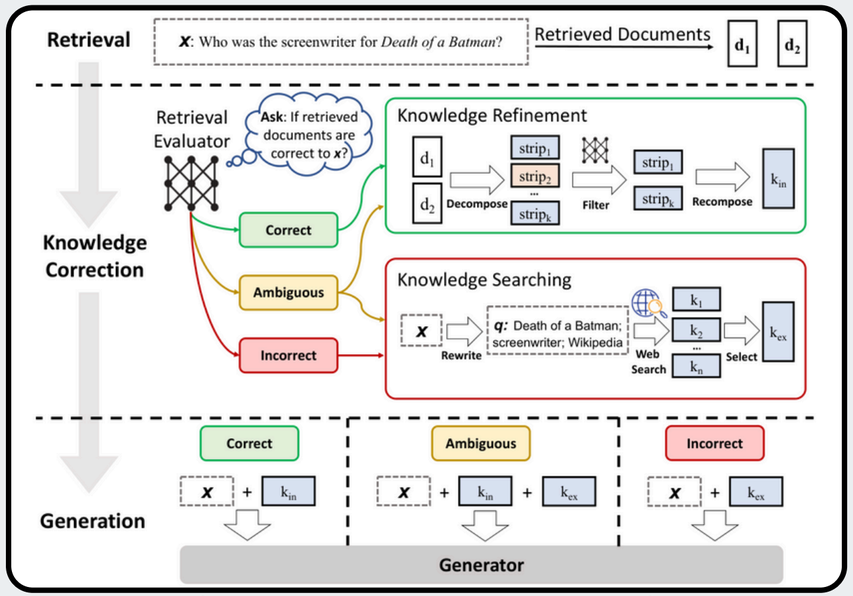
$$P(Y|X) = P(D|X)P(Y, D|X)$$

P(Y|X): 완벽한 답변(Y)를 내놓을 확률
P(D|X): 완벽한 문서(D)가 검색될 확률
P(Y, D|X): 완벽한 문서(D)가 검색되고 완벽한 답변(Y)이 생성될 결합 확률

대부분의 기존 RAG 방식은 검색된 문서의 품질을 확인하지 않고 무분별하게 통합(indiscriminately incorporate)하여 사용함.*

RAG가 Hallucination 줄이기 위한 기술임에도 불구하고, 역설적으로 Retriever(검색기) 가 Hallucination을 **악화시키는 원인**이 될 수 있음.

1. 연구배경 및 문제정의: CRAG의 제안



(CRAG 모델의 추론 동작 개요)

이를 해결하기 위해 논문은 **CRAG (Corrective Retrieval Augmented Generation)**를 제안함.

기존의 RAG에 스스로 정보를 평가하고 **교정하는(Corrective)** 개념을 더함.

검색이 실패하는 시나리오에 집중하여, 잘못된 정보에 흔들리지 않는
생성의 **Robustness(강건성)** 을 향상시키는 것을 목표로 함.

CRAG 추론(inference) 흐름

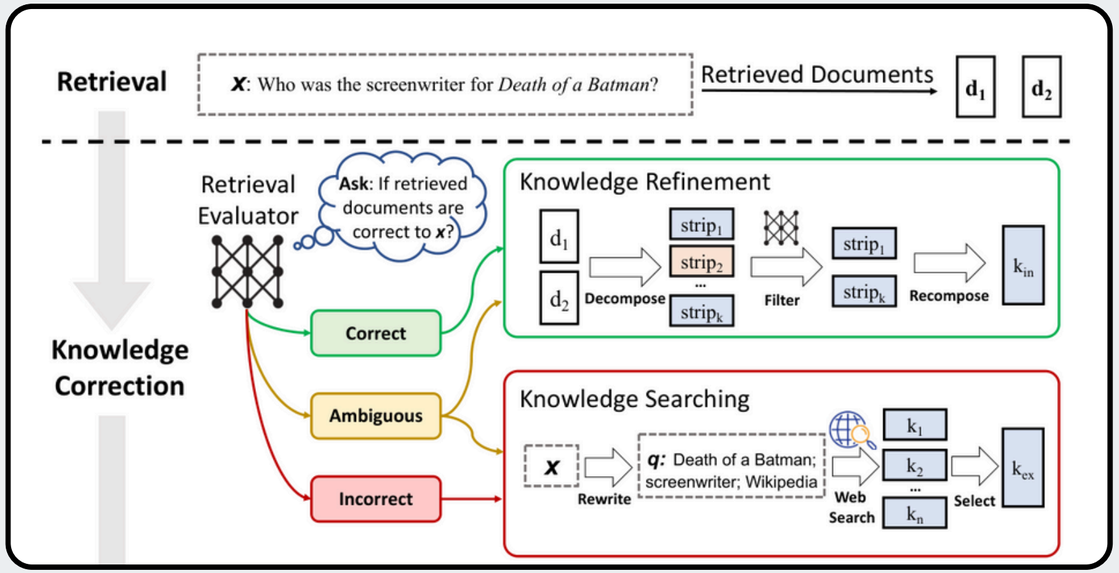
1. 평가 (Evaluate): 먼저, 경량 검색 평가기**(Retrieval Evaluator)**가 검색된 정보의 신뢰도를 평가.

2. **Action Trigger**(행동 촉발): 평가 점수를 바탕으로 {Correct, Incorrect, Ambiguous} 세 가지 행동 중 하나를 **촉발(trigger)**할지 결정.

3. 교정 (Correct): 촉발된 행동에 따라, 지식 정제**(Knowledge Refinement)** 또는 웹 검색**(Web Search)**과 같은 맞춤형 교정 작업을 수행.

4. 생성 (Generate): 마지막으로, 교정된 고품질의 정보를 바탕으로 최종 답변을 생성.

2. 핵심 아이디어: Retrieval Evaluator



(CRAG 모델의 검색 및 지식 교정 단계)

	Accuracy
Our Retrieval Evaluator (T5-based)	84.3
ChatGPT	58.0
ChatGPT-CoT	62.4
ChatGPT-few-shot	64.7

(CRAG 평가기와 ChatGPT의 검색 정확도 비교)

Retrieval Evaluator의 역할

- Generator에 정보를 넣기 전, 검색된 각 문서가 사용자의 질문에 얼마나 관련 있고 정확한지 Confidence score(신뢰도 점수)를 평가함.

- 평가 점수를 기반으로 쓸모있는 정보와 버려야 할 정보를 구분하여, 부정확한 정보로 인한 LLM의 Hallucination 현상을 방지하는 핵심적인 역할을 수행함.

특징

- 경량(Lightweight) 모델 사용.

- 거대한 LLM 대신, T5-large (0.77B) 모델을 Fine-tuning하여 사용함.

- 이는 Self-RAG의 평가 모델(Critic)인 LLaMA-2 (7B)보다 약 10배 작고 가벼워, 훨씬 적은 비용으로 빠르고 효율적인 평가가 가능함.

작동 방식

- 질문과 검색된 문서 1개를 쌍으로 입력받아, -1 (관련성이 낮음)부터 +1 (관련성이 높음) 사이의 Relevance Score(관련성 점수)를 개별적으로 계산함.

- 이 점수들을 바탕으로 사전에 정의된 임계값(threshold)에 따라 다음 행동(Correct, Incorrect, Ambiguous)을 촉발(trigger)함.

2. 핵심 아이디어: Retrieval Evaluator - fine-tuning

```
Who was the producer of Gladiator? [SEP] Ferraro financed, developed and packaged the... launch a live "American Gladiators" show on the    0
Who was the producer of Gladiator? [SEP] Gladiator (2000 film) Gladiator is a 2000 epic historical drama film ..., Maximus rises through the ranks of the gladiatorial  1
```

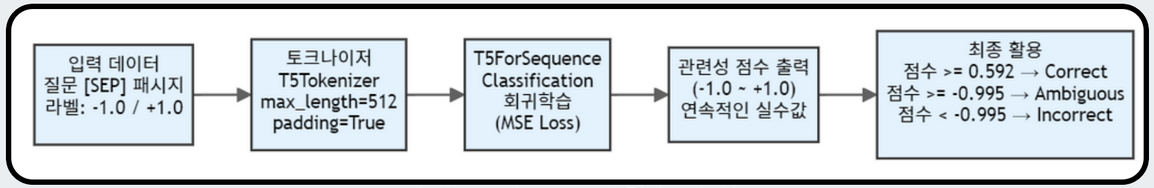
(PopQA 기반 T5 Evaluator 학습 데이터 예시)

```
# 실제 변환 코드 (train_evaluator.py)
label.append((int(l.strip()) - 0.5) * 2)
# 변환 결과:
# 0 → (0 - 0.5) * 2 = -1.0
# 1 → (1 - 0.5) * 2 = +1.0
```

(파일에서는 0과 1로 저장되어 있지만, 학습할 때는
이 공식을 통해 -1과 +1로 변환됨.)

```
{
  "question": "What is George Rankin's occupation?",
  "s_wiki_title": "George Rankin",
  "ctxs": [
    {"title": "George Rankin", "text": "George James Rankin... politician"},
    {"title": "John Smith", "text": "John Smith was a lawyer..."},
    {"title": "Jane Doe", "text": "Jane Doe worked as..."},
    // ... 총 10개 (이미 검색된 결과)
  ]
}
```

(PopQA 원본 데이터의 JSON 구조)



(T5 Evaluator의 학습 과정)

학습 모델: T5-large (0.77B)

학습 데이터셋: PopQA[1]

- Popularity-based QA - Wikipedia 인기도 기반의 데이터 셋, Longtail 데이터 셋
- Retrieval Evaluator를 학습시키는 데 사용된 유일한 데이터셋
- 총 14,000개의 샘플 중, 테스트에 사용된 1,399개를 제외한 나머지가 정보 유출을 방지하기 위해 학습에 사용됨.

Evaluator는 관련성 정도를 예측하는 **회귀(Regression)** 문제로 학습됨.

Positive 샘플 (Label: +1):

- PopQA 데이터셋의 검색 결과 중에서 패시지 제목이 Golden Subject Wiki Title(정답 위키 제목) 과 일치하는 문서를 긍정 샘플로 사용함.

Negative 샘플 (Label: -1):

- 검색 결과 중에서, 패시지 제목이 정답 위키 제목과 일치하지 않는 나머지 모든 문서(논문에서는 무작위)부정 샘플로 사용함.

[1] akariasai/PopQA · Datasets at Hugging Face

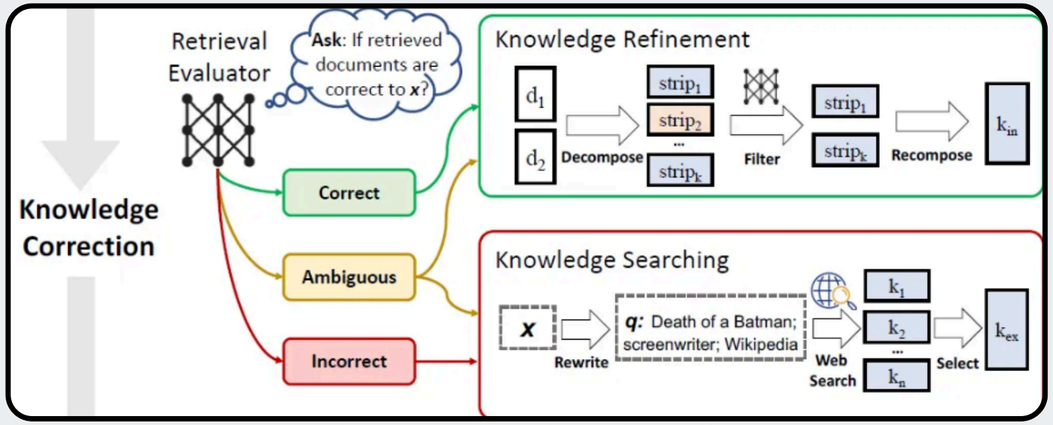
2. 핵심 아이디어 - Action Trigger

Algorithm 1: CRAG Inference

Require : E (Retrieval Evaluator), W (Query Rewriter), G (Generator)
Input : x (Input question), $D = \{d_1, d_2, ..., d_k\}$ (Retrieved documents)
Output : y (Generated response)

$score_i$ = E evaluates the relevance of each pair (x, d_i), $d_i \in D$
Confidence = Calculate and give a final judgment based on $\{score_1, score_2, ...score_k\}$
`// Confidence has 3 optional values: [CORRECT], [INCORRECT] or [AMBIGUOUS]`
if *Confidence* == `[CORRECT]` **then**
    Internal_Knowledge = Knowledge_Refine(x, D)
    k = Internal_Knowledge
else if *Confidence* == `[INCORRECT]` **then**
    External_Knowledge = Web_Search(W Rewrites x for searching)
    k = External_Knowledge
else if *Confidence* == `[AMBIGUOUS]` **then**
    Internal_Knowledge = Knowledge_Refine(x, D)
    External_Knowledge = Web_Search(W Rewrites x for searching)
    k = Internal_Knowledge + External_Knowledge
end
G predicts y given x and k

(CRAG 추론 알고리즘)



(CRAG 모델의 교정 단계)

Correct (정확)

- 검색된 문서 중 하나라도 신뢰도 점수가 설정된 **상한 임계값**(예: PopQA (**0.59**, -0.99))을 넘으면 **Correct** 행동이 실행됨.
- 이는 검색 결과에 신뢰할 수 있는 관련 정보가 포함되어 있음을 의미함. 다만, 관련 문서 내에도 불필요한 정보가 있을 수 있으므로,
- 지식 정제(Knowledge Refinement) 과정을 통해 핵심적인 정보만 추출하여 generator(생성모델)에 사용함.

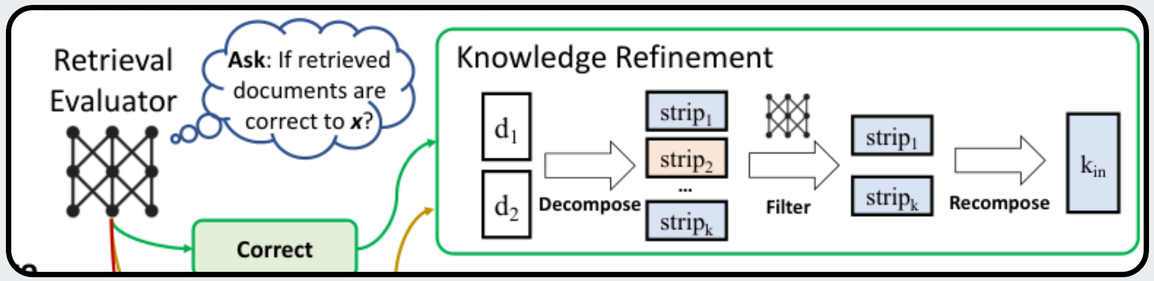
Incorrect (부정확)

- 검색된 모든 문서의 신뢰도 점수가 **하한 임계값**(예: PopQA (0.59, **-0.99**))보다 낮으면 **Incorrect** 행동이 실행됨.
- 이는 모든 검색 결과가 질의와 관련이 없어 생성에 도움이 되지 않음을 나타냄.
- 잘못된 정보에 기반하여 조작된 사실을 생성하는 것을 막기 위해, 기존 문서는 폐기하고 **웹 검색(Web Search)**을 통해 새로운 외부 지식을 탐색하고 교정을 시도함.

Ambiguous (모호)

- 검색 결과의 정확성을 명확히 판단하기 어려워 평가 점수가 **상한과 하한 임계값** (예: PopQA (0.59, -0.99))사이에 위치할 경우 Ambiguous 행동이 실행됨.
- 이는 평가기가 스스로의 판단을 확신하지 못하는 상태를 의미함.
- 이러한 불확실성에 대응하기 위해, Correct와 Incorrect의 처리 방식을 모두 사용함.
- 즉, **정제된 내부 지식**과 **웹 검색**을 통한 외부 지식을 결합하여 서로 보완함으로써 시스템의 robustness(강건성)과 resilience(회복력)을 강화함.

2. 핵심 아이디어 - Knowledge Refinement



(CRAG 모델의 Knowledge Refinement 단계)

검색된 문서가 질문과 Correct로 판단되더라도, 문서 전체에는 불필요한 noise가 포함될 수 있음

Knowledge Refinement(지식 정제)는 이러한 관련 문서 내에서 가장 핵심적인 정보만을 정밀하게 추출하여 Generator의 정확도를 높이는 과정.

Knowledge Refinement(지식정제) 프로세스

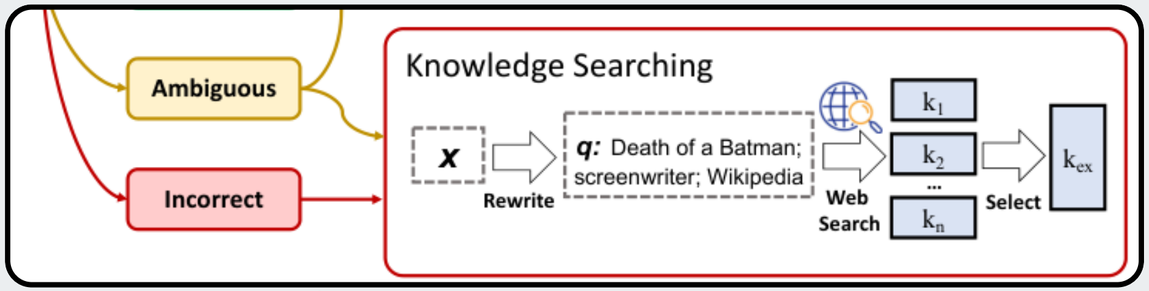
Decompose-then-Recompose(분해-재구성)

1. Decompose(분해)
   - 관련성이 확인된 문서(d)를 문장 단위의 작은 strip(지식조각)으로 분할함.
   - 한두 문장의 짧은 문서는 그 자체로 하나의 조각이 되며, 긴 문서는 여러 조각으로 나뉨. 각 조각은 독립적인 정보를 담고 있는 것으로 간주함.

2. Filter(필터링)
   - 앞서 사용된 Retrieval Evaluator(검색 평가기)를 각 지식 조각에 재적용하여, 원래 질문과의 관련성 점수를 개별적으로 다시 계산합니다.
   - 이 점수를 기반으로 관련성이 낮은 불필요한 조각들을 걸러냅니다.

3. Recompose(재구성)
   - 필터링을 통과한, 관련성 높은 지식 조각들만 원래 순서대로 다시 연결(concatenate)함.
   - 이렇게 완성된 최종 결과물이 바로 생성 모델에 전달될 핵심적인 Internal Knowledge(내부지식, k_{in})

2. 핵심 아이디어 - Web Search



(CRAG 모델의 Web Search 단계)

기존에 가지고 있는 Internal Knowledge(내부 지식)만으로는 답변이 불가능하거나, 검색된 문서들이 모두 질문과 관련이 없다고(Incorrect) 판단될 때, hallucination 현상을 방지하고 더 정확한 답변을 생성하기 위해 External Knowledge(외부 지식) 소스로 웹을 활용함.

Knowledge Searching(지식검색) 프로세스

1. Query Rewriting(질의 재작성)
   - 사용자의 원래 질문(x)을 실제 검색 엔진 사용 패턴처럼 키워드 중심의 검색어(q')로 재작성함.
   - 논문에서는 이 과정을 위해 ChatGPT를 활용함.

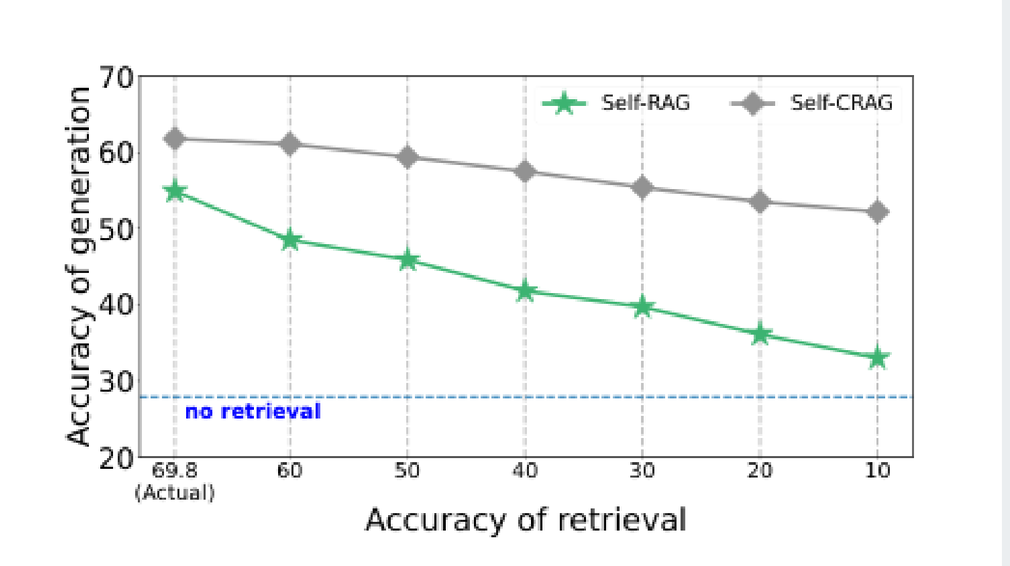
2. Web Search(웹 검색)
   - 재작성된 검색어를 사용해 상용 웹 검색 API(논문에서는 Google Search API 사용)를 호출하여 관련성이 높은 웹 페이지들의 URL 목록을 얻음.

3. Select & Refine(선택 및 정제)
   - 웹 정보의 신뢰도를 높이기 위해 위키피디아(Wikipedia)와 같이 권위 있는 소스를 우선적으로 고려함.
   - 획득한 웹 페이지 내용에 Knowledge Refinement(지식 정제)와 동일한 방식을 적용하여 질문 답변에 필요한 핵심 정보만 추출함.
   - 이렇게 최종적으로 정제된 정보를 External Knowledge(외부지식, k_{ex}) 이라고 하며, 이를 generator에 전달함.

3. Experimental Results

Method	PopQA (Accuracy)	Bio (FactScore)	Pub (Accuracy)	ARC (Accuracy)
LMs trained with propriety data				
LLaMA2-c$_{13B}$	20.0	55.9	49.4	38.4
Ret-LLaMA2-c$_{13B}$	51.8	79.9	52.1	37.9
ChatGPT	29.3	71.8	70.1	**75.3**
Ret-ChatGPT	50.8	-	54.7	**75.3**
Perplexity.ai	-	71.2	-	-
Baselines without retrieval				
LLaMA2$_{7B}$	14.7	44.5	34.2	21.8
Alpaca$_{7B}$	23.6	45.8	49.8	45.0
LLaMA2$_{13B}$	14.7	53.4	29.4	29.4
Alpaca$_{13B}$	24.4	50.2	55.5	54.9
CoVE$_{65B}$	-	71.2	-	-
Baselines with retrieval				
LLaMA2$_{7B}$	38.2	78.0	30.0	48.0
Alpaca$_{7B}$	46.7	76.6	40.2	48.0
SAIL	-	-	69.2	48.4
LLaMA2$_{13B}$	45.7	77.5	30.2	26.0
Alpaca$_{13B}$	46.1	77.7	51.1	57.6
LLaMA2-hf-7b				
RAG	50.5	44.9	48.9	43.4
CRAG	54.9	47.7	59.5	53.7
Self-RAG*	29.0	32.2	0.7	23.9
Self-CRAG	49.0	69.1	0.6	27.9
SelfRAG-LLaMA2-7b				
RAG	52.8	59.2	39.0	53.2
CRAG	59.8	74.1	**75.6**	68.6
Self-RAG	54.9	81.2	72.4	67.3
Self-CRAG	**61.8**	**86.2**	74.8	67.2

(a). 4개 데이터셋의 테스트 세트에 대한 전반적인 평가 결과



(b). SelfRAG-LLaMA-7b를 사용한 PopQA 데이터셋에서, 검색 성능변화에 따른 Self-RAG와 Self-CRAG의 답변 생성 성능에 대한 전반적인 평가 결과

3. Experimental Results

	LLaMA2-hf-7b	SelfRAG-LLaMA2-7b
CRAG	54.9	59.8
w/o. Correct	53.2	58.3
w/o. Incorrect	54.4	59.5
w/o. Ambiguous	54.0	59.0
Self-CRAG	49.0	61.8
w/o. Correct	43.6	59.6
w/o. Incorrect	47.7	60.8
w/o. Ambiguous	48.1	61.5

(c). 정확도 관점에서 PopQA 데이터셋에 대한 각 단일 행동 제거 연구.

	LLaMA2-hf-7b	SelfRAG-LLaMA2-7b
CRAG	54.9	59.8
w/o. refinement	49.8	54.2
w/o. rewriting	51.7	56.2
w/o. selection	50.9	58.6
Self-CRAG	49.0	61.8
w/o. refinement	35.9	52.2
w/o. rewriting	37.2	58.4
w/o. selection	24.9	57.9

(d). 정확도 관점에서 PopQA 데이터셋에 대한 각 지식 활용 연산 제거 연구

3. Experimental Results

	LLaMA2-hf-7b	SelfRAG-LLaMA2-7b
PopQA		
CRAG	54.9	59.8
RAG	50.5	52.8
RAG w. web	52.2	53.8
Self-CRAG	49.0	61.8
Self-RAG	29.0	54.9
Self-RAG w. web	24.9	57.9

(e).동일한 입력에 대한 CRAG, Self-CRAG와 RAG, Self-RAG의 정확도 비교 결과.

	TFLOPs per token	executing time(s)
RAG	26.5	0.363
CRAG	27.2	0.512
Self-RAG	26.5∼132.4	0.741
Self-CRAG	27.2∼80.2	0.908

(f). GPU에서의 토큰 당 FLOPs와 인스턴스 당 실행 시간에 대한 RAG, CRAG, Self-CRAG, Self-RAG의 계산 오버헤드 평가

4. Conclusion

1. 문제 정의

- 기존 RAG는 검색(Retrieval) 결과가 부정확할 경우, 생성 모델에 잘못된 정보를 그대로 전달하여 답변의 신뢰도를 떨어뜨리는 본질적인 한계를 가짐.

2. 해결 방안 (CRAG)

- 이 문제를 해결하기 위해, 검색된 정보의 품질을 먼저 평가하는 Retrieval Evaluator(검색 평가기)를 도입함.
- 평가 결과에 따라 Correct, Incorrect, Ambiguous 세 가지 행동을 다르게 수행하며, 웹 검색과 지식 정제/선택 같은 최적화된 활용법을 통해 자동 교정 능력을 극대화함.

3. 핵심 장점 (확장성)

- CRAG는 plug-and-play 방식으로 설계되어, 기존의 다양한 RAG 시스템에 손쉽게 결합할 수 있는 높은 확장성을 가짐.
- 이는 CRAG가 독립적인 성능 향상 모듈처럼 작동할 수 있음을 의미함.

4. 명확한 한계점 (의존성 이전)

- CRAG는 기존 Retriever에 대한 의존도를 낮추는 대신, 새로운 Retrieval Evaluator에 대한 의존성을 만들어냄.
- 이 평가기 모델은 별도의 미세 조정(fine-tuning) 과정이 반드시 필요하다는 한계를 가짐.

5. Open Question

1. CRAG 구현후 테스트

- 실제 RAG 와 Web Search API 적용

2. 생성정확도 향상을 위한 ‘검색 전 판단’ 아이디어 모델과 비교

- [1]Adaptive-RAG

[1] [2403.14403] Adaptive-RAG: Learning to Adapt Retrieval-Augmented Large Language Models through Question Complexity

부록1. 유사 모델 비교 분석 - 교정(Correction) 중심으로

Self-Rag[1]

- LLM이 검색을 수행한 뒤 답변을 생성하며, 동시에 [관련성], [근거] 등을 나타내는 Critique Token(비평토큰)을 생성함.
- 이 평가 점수를 기반으로 가장 품질이 좋은 답변을 선택하거나, 관련 없는 정보는 무시하는 방식으로 교정/처리함.

Plan-Rag[2]

- 수립된 계획에 따라 데이터를 검색하고, 그 결과를 Observation(관찰)로 받음. 이후, 이 관찰 내용이 현재 계획을 진행하기에 부적합하다고 판단되면 Re-plan(Re-plan)을 통해
- 계획 자체를 수정하여 다음 행동을 바로잡음.

Rag-star[3]

- 모델이 내부 지식으로 먼저 추론 경로를 생성하면, 각 추론 단계마다 외부 문서를 검색함.
- 검색된 결과를 바탕으로 Reward Model(보상모델)이 추론의 타당성을 평가하고, 만약 모델의 생각과 외부 정보가 충돌하면 검색된 내용을 근거로 추론 경로를 수정함.

[1] [2310.11511] Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection
[2] [2406.12430] PlanRAG: A Plan-then-Retrieval Augmented Generation for Generative Large Language Models as Decision Makers
[3] [2412.12881] RAG-Star: Enhancing Deliberative Reasoning with Retrieval Augmented Verification and Refinement