Generate-then-Ground in Retrieval-Augmented Generation for Multi-hop Question Answering

Announcement Date	@2025년 2월 6일
Conference Name	ACL 2024
Keywords	LLM Mulit-Hop RAG

🔷 멀티 홉 검색(Multi-Hop Retrieval)이란?

기본적인 검색 방식은 하나의 쿼리(Query)에 대해 단일 문서(Document) 또는 단일 소스에서 답을 찾는 방식입니다. 그러나 **멀티 홉 검색**은 **여러 개의 문서를 연결하여 추론**해야 할 때 사용됩니다.

➡️ 즉, 단일 문서로는 충분한 정보를 얻을 수 없을 때, 여러 문서를 넘나들며 관련 정보를 조합하여 답을 찾는 방식입니다.

🔷 멀티 홉의 예시

1️⃣ 단일 홉(Single-Hop) vs 멀티 홉(Multi-Hop)

- **단일 홉 검색** (Single-Hop Retrieval)

  질문: "에디슨은 언제 태어났어?"

  - 검색 엔진이 한 번의 검색으로 에디슨의 출생 정보를 포함한 문서를 찾고, 답을 생성함.
  - 하나의 문서에서 답을 찾을 수 있음.
  - ✔️ *정답:* "1847년 2월 11일"

- **멀티 홉 검색** (Multi-Hop Retrieval)

  질문: "에디슨이 태어난 해에 미국 대통령은 누구였어?"

  - 1단계: "에디슨은 1847년에 태어났다."

- 2단계: "1847년 미국 대통령은 제임스 K. 포크였다."
- 여러 문서를 연결하여 추론해야 함.
- ✔️ *정답*: "제임스 K. 포크"

➡️ 이처럼 하나의 정보만으로는 답을 도출할 수 없는 경우, 여러 문서를 참조하여 추론하는 것이 멀티 홉입니다.

Introduction

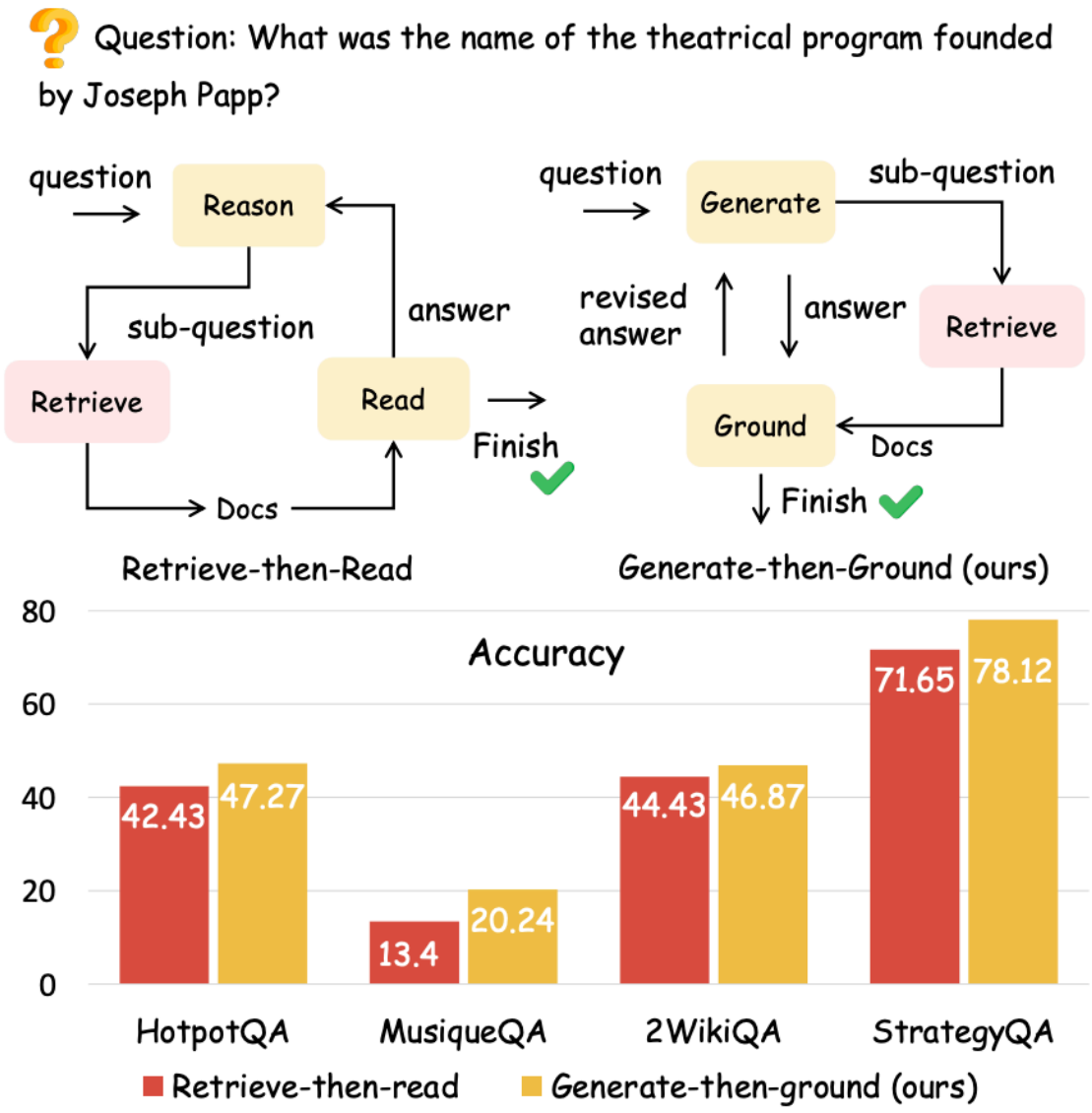

Figure 1: The top block depicts the comparison with the commonly-used *retrieve-then-read* paradigm in MHQA task. The bottom block provides the performance of our method and baselines in four MHQA benchmarks.

- 기존 다단계 질문 응답 방식인 **검색 후 읽기(Retrieve-then-Read)** 는 검색기의 한계와 문서 노이즈 문제로 인해 성능이 저하되는 문제가 있었음.
    - 검색기의 성능 제한 : 검색된 문서 내에서만 답변을 찾아야 하므로, LLM의 내제된 지식을 충분히 활용 X

    - 문서 노이즈 : 검색된 문서에 관련 없는 정보나 그럴듯하지만 잘못된 정보가 포함될 가능성이 있음
- 이를 해결하기 위해, 우리는 **생성 후 검증(Generate-then-Ground, GenGround)** 프레임워크를 제안했음.
- GenGround는 **LLM이 먼저 답변을 생성한 후, 검색된 문서를 활용해 검증하고 필요하면 수정하는 방식**으로 동작했음.
- 작은 모델에서도 이 방식을 사용할 수 있도록 **지식 증류(Instructional Grounding Distillation) 기법**을 도입했음.
- 4개의 데이터셋에서 실험한 결과, **GenGround가 기존 방법보다 우수한 성능을 기록했음.**

Generate-then-Ground with LLM

Figure 2에서 볼 수 있듯이, GenGround는 LLM이 두 가지 단계를 반복하도록 하여 최종 답변을 도출함

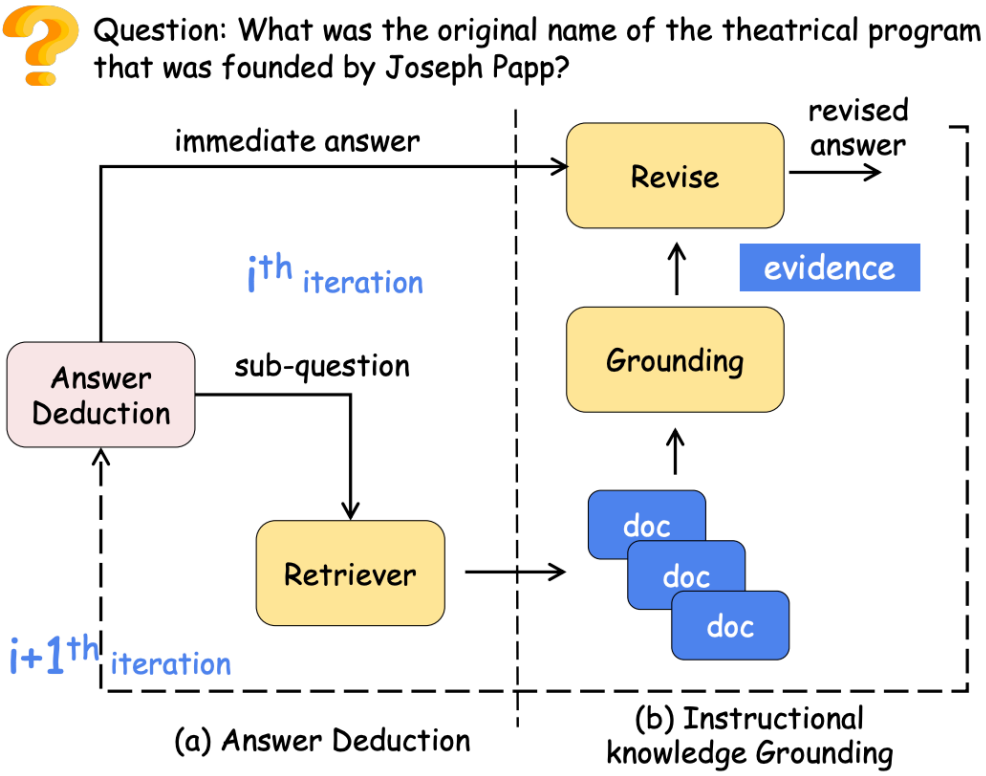

Figure 2: The architecture of the proposed generate-then-ground framework.

1. 답변 생성(Answer Deduction) 단계
    - LLM이 입력 질문을 바탕으로 더 단순한 **단일 단계 질문**을 만들고 즉각적인 답변을 생성
2. 지식 검증 (Instructional Knowledge Grounding) 단계
    - 검색된 문서를 활용하여 질문-답변 쌍을 검토하고 필요한 경우 수정

Generalization with Grounding Distillation

- 대규모 언어 모델(LLM)은 지식 검증 능력이 뛰어나지만, 실행 비용이 크고 속도가 느리다는 문제가 있음.
- 이를 해결하기 위해, 우리는 지식 검증 증류(IGD) 기법을 도입하여 작은 모델도 LLM의 검증 과정을 학습할 수 있도록 함.
- Natural Questions(NQ) 데이터셋에서 50,000개 질문을 샘플링하고, ChatGPT를 활용해 검증 및 수정 과정을 데이터셋으로 구축함.

- 작은 모델이 검색된 문서에서 증거를 찾고, 기존 답변을 수정하는 과정을 학습하도록 최적화함.
- 이 방법을 통해 작은 모델도 LLM 수준의 검증 능력을 갖출 수 있도록 일반화함.

Experimental Setup

- Dataset
  - 네 가지 Multi-Hop QA (MHQA) 데이터셋을 사용하여 실험 진행
- Baselines
  1. 검색 없이 생성하는 방법 (Generation w/o Retrieval)
     - **CoT (Wei et al., 2022)**: LLM이 문제를 해결할 때, 단계별 중간 추론 과정을 포함하여 답변을 생성하는 기법.
     - **CoT-SC (Wang et al., 2022b)**: 다양한 추론 경로를 생성한 후, 가장 일관된 답변을 선택하는 기법.
     - **GenRead (Yu et al., 2022)**: LLM이 검색된 문서 없이 자체적으로 문맥을 생성한 후 답변을 생성하는 방식.
     - **GenRead + 분해 (w/ decomposition)**: 다단계 질문을 먼저 단일 단계 질문으로 나누고, 각각을 GenRead로 해결하는 방식.
  2. 검색을 활용하는 방법 (Gereration w/ Retrieval)
     - **VE (Zhao et al., 2023)**: 검색된 문서를 검증하고 수정하는 방식.
     - **ReAct (Yao et al., 2022)**: 질문 생성, 문서 검색, 지식 활용 과정을 결합하여 답변을 생성하는 기법.
     - **GRG + 분해 (GRG w/ decomposition)**: 검색된 문서와 LLM이 자체적으로 생성한 문서를 함께 활용하여 답변을 생성하는 기법.
     - **RetGen (Shao et al., 2023)**: 검색과 생성 과정을 반복적으로 수행하여 답변을 생성하는 방식.
     - **DSPy (Khattab et al., 2023)**: LLM을 기반으로 한 프로그래밍 프레임워크를 활용한 검색 증강 기법.
     - **SearChain (Xu et al., 2024)**: 검색기와 상호작용하여 생성된 답변을 검증하고 수정하는 방식.
- Implementation Details
  - LLM Backbone : GPT-3.5-turbo

- Batch = 3
- Retriever : ColBERTv2 (검색된 문서의 ranking 정하는 역할), BM25, Google Search
- 지식 검증 증류(IGD) 모델 학습:
  - Mistral-7B 모델을 50K 개의 데이터로 학습 진행.
  - deepspeed ZeRO 전략(Rasley et al., 2020)을 사용하여 학습 성능 최적화.
  - 학습 속도: NVIDIA A100-PCIE-80GB GPU 3개를 사용하여 18시간 내 학습 완료.

Experimental Results

Methods	HotpotQA			MuSiQue			2Wikimultihopqa			StrategyQA
	F1	Acc	Acc†	F1	Acc	Acc†	F1	Acc	Acc†	Acc
Generate w/o Retrieval										
CoT (Wei et al., 2022)	35.28	30.79	37.07	23.35	13.21	17.85	35.41	32.46	34.52	67.83
CoT-SC (Wang et al., 2022b)	42.25	38.68	39.07	15.61	10.02	12.42	40.37	36.57	38.59	70.84
GenRead (Yu et al., 2022)	35.21	36.81	37.54	9.77	9.29	10.32	23.13	20.62	28.31	67.13
GenRead *w/ decomposition*	42.28	43.32	45.31	20.13	17.58	20.62	41.19	41.63	43.24	68.13
Generate w/ Retrieval										
VE (Zhao et al., 2023)	29.64	22.64	24.64	6.5	11.14	15.57	13.76	31.57	32.64	63.07
ReAct (Yao et al., 2022)	40.70	33.10	37.12	15.34	17.32	19.32	35.50	30.10	33.41	68.37
GRG *w/ decomposition*	50.21	45.18	50.80	24.87	17.91	22.33	40.42	40.48	43.05	75.21
RetGen (Shao et al., 2023)	28.30	41.04	44.10	21.04	17.69	20.19	36.00	42.17	45.21	73.42
SearChain (Xu et al., 2024)	-	46.76	48.12	-	17.07	20.45	-	42.14	46.27	76.95
DSPy (Khattab et al., 2023)	47.80	42.43	50.07	20.11	13.40	17.40	44.77	43.43	45.43	71.78
GenGround (Ours)	**52.26**	**47.27**	**55.73**	**27.36**	**20.24**	**24.77**	**50.21**	**45.61**	**48.58**	**77.12**

Table 2: Evaluation results on multi-hop question answering datasets. Acc† indicates the semantic accuracy of model outputs evaluated with `gpt-3.5-turbo-instruct` with the same prompt. Since the *SearChain* prompts the LLM to generate a long-form answer while the ground truth answer in our dataset is short-form, we only evaluate it with the Acc and Acc† metrics.

Method	HQA	MQA	WQA	Average Δ ↓
Retriever → BM25				
Ours	42.21	18.32	40.32	-
DSPy	40.86	15.32	30.85	5.27↓
GRG *w/ dq*	41.31	15.62	38.84	2.36↓
RetGen	39.12	8.41	35.83	6.50↓
SearChain	39.57	14.93	37.41	3.65↓
Retriever → Google Search				
Ours	48.95	21.54	46.87	-
DSPy	46.86	20.71	39.92	3.29↓
GRG *w/ dq*	42.57	18.41	43.21	4.39↓
RetGen	42.82	14.27	44.31	5.32↓
SearChain	44.35	19.76	44.39	2.95↓

Table 5: Accuracy (Acc) on three datasets using BM25 and Google Search as retrievers, respectively. The *w/ dq* is short for *without decomposition*.

- GenGround는 다단계 질문 응답(MHQA) 데이터셋에서 모든 기존 방법보다 높은 정확도를 기록했음.
- Mistral-7B와 같은 작은 모델에서도 지식 검증 증류(IGD) 기법을 적용하면 성능이 크게 향상됨.
- 다양한 검색기(BM25, Google Search)에서도 성능이 우수하며, 특히 고품질 검색 결과를 활용할 때 성능이 극대화됨.
- 답변 생성, 지식 검증, 배치 검증 등의 주요 기법이 각각 성능 향상에 필수적인 요소임을 실험을 통해 입증했음.
- 사례 연구에서도 GenGround는 기존 방법보다 신뢰할 수 있는 답변을 생성하며, 잘못된 정보를 수정하는 능력이 뛰어남을 확인했음.

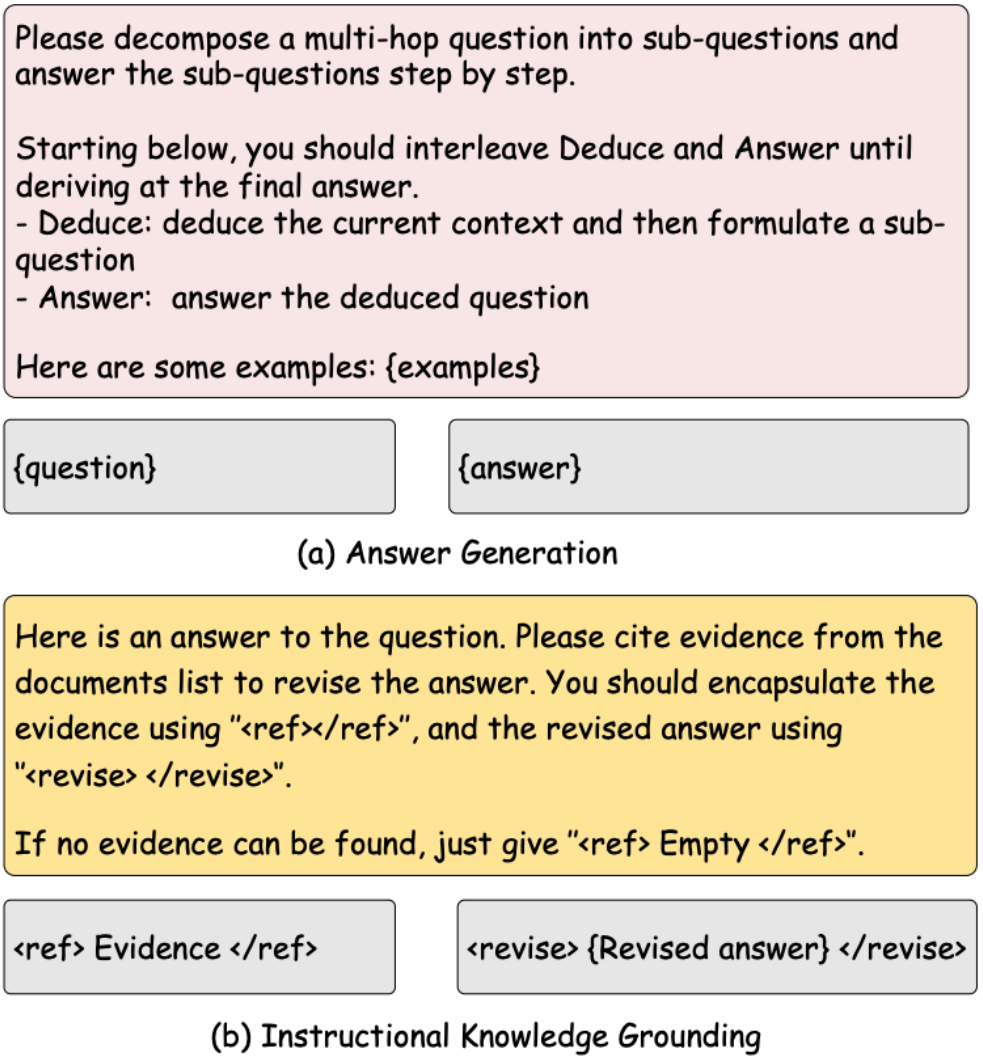

Figure 3: The instruction for the *answer deduction* (a) and *instructional knowledge grounding*(b) phases in our framework. The pink and yellow blacks indicate the input while the gray blocks indicate the output.

Limitation

- GenGround는 초기 답변 생성이 필수적이므로, 특정 질문에서는 의미 없는 답변이 생성될 가능성이 있음.
- 모든 복잡한 질문이 단순하게 변환될 수 있는 것은 아니며, 정보 손실이 발생할 수 있음.

- 검색된 문서가 신뢰할 수 없는 경우, 모델이 잘못된 답변을 생성할 가능성이 있어 추가적인 문서 평가 방법이 필요함.