복잡성에서 원자성으로: 지식 기반 이중 재작성 및 추론을 통한 증강 생성 능력 향상

왕진유(Jinyu Wang)[*1] **푸징징**(Jingjing Fu)[*1] **왕루이**(Rui Wang)[1] **송레이**(Lei Song)[1] **비안장**(Jiang Bian)[1]

초록

검색 강화 생성(RAG) 시스템의 최근 발전은 외부 지식 검색을 통합함으로써 대규모 언어 모델(LLM)의 역량을 크게 향상시켰습니다. 그러나 검색에만 의존하는 방식은 심층적이고 전문적인 지식을 발굴하고, 특정 분야의 복잡한 질문을 해결하는 데 필요한 논리적 추론을 수행하는 데 종종 부적절합니다. 이러한 과제를 해결하기 위해, 우리는 전문 지식을 원자적 방식으로 추출·이해·활용하면서 동시에 일관된 추론 근거를 구축하도록 설계된 접근법을 제시한다. 이 접근법의 핵심에는 네 가지 중추적 구성 요소가 있습니다: 원시 데이터에서 원자적 태그를 추출하는 지식 원자화기, 초기 질의를 촉진하기 위해 후속 질문을 생성하는 질의 제안기, 원자적 지식 정렬을 기반으로 지식을 찾는 원자적 검색기, 그리고 검색된 정보를 바탕으로 질의할 원자적 태그와 청크 쌍을 결정하는 원자적 선택기입니다. 이러한 접근 방식을 통해, 우리는 지식 인식 작업 분해 전략을 구현하여 초기 질문과 획득한 지식에 부합하는 논리를 반복적으로 구축합니다. 다양한 벤치마크, 특히 다단계 추론이 필요한 경우에서 본 접근법의 효용성을 입증하기 위한 포괄적인 실험을 수행했습니다. 두 번째로 우수한 방법 대비 최대 +10.1(20.4%)에 달하는 상당한 성능 향상은 복잡하고 지식 집약적인 응용 분야에서 본 접근법의 잠재력을 강조합니다. 코드는 https://github.com/microsoft/PIKE-RAG에서 공개적으로 이용 가능합니다.

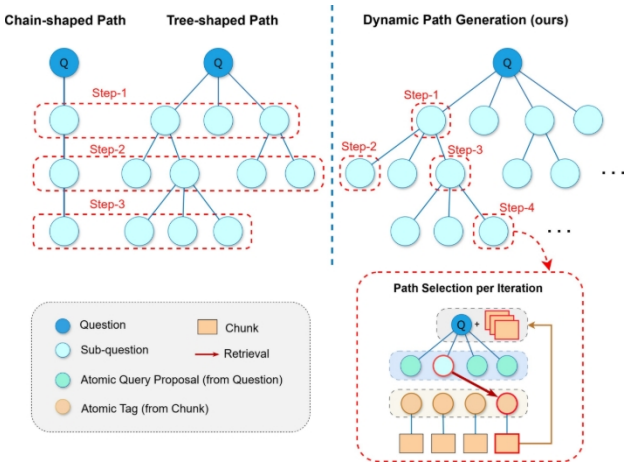


그림 1. 복잡한 질문은 일반적으로 사슬형 또는 트리형 경로를 따라 하위 질문으로 분해되며, 해결을 위한 관련 정보 수집에는 청크 검색이 활용된다. 기존 접근법과 달리, 본 방법은 질문과 청크에 대한 이중 재작성을 통한 원자적 지식 정렬로 질문 분해와 정보 검색을 원활하게 통합하고, 원자적 쌍 검색 및 선택을 통해 후속 하위 질문을 동적으로 결정함으로써 검색된 지식에 기반하여 진화하는 적응적이고 상호작용적인 분해 경로를 가능하게 한다.

1. 서론

대규모 언어 모델(LLMs)은 일관성 있고 문맥에 부합하는 텍스트를 생성하는 능력과 텍스트 완성부터 번역 및 요약에 이르기까지 다양한 언어 작업을 수행할 수 있는 다재다능함을 보여주며 자연어 처리 분야에 혁명을 일으켰다(Achiam et al., 2023; Touvron et al., 2023). 광범위한 능력에도 불구하고, LLM은 전문 분야의 특수한 질의에 대해 뚜렷한 한계를 보입니다(Ling et al., 2024; Wang et al., 2023a). 이는 주로 도메인별 훈련 자료(예: *미공개 문서*)의 부족과 해당 도메인 내 전문 지식 및 논리(예: *산업별 약어, 기업별 운영 규칙*)에 대한 불완전한 이해에서 기인합니다. 그 결과, LLM은 잠재적으로 오류가 있을 뿐만 아니라 전문가 수준의 참여에 필요한 세부 사항과 정확성이 부족한 응답을 생성할 수 있습니다 (Bender 외, 2021).

[*]동등한 기여 [1]마이크로소프트 리서치 아시아, 중국 베이징. 연락처: Lei Song <lesong@microsoft.com >.

제42회 국제 기계 학습 학회 논문집, 캐나다 밴쿠버. PMLR 267, 2025.

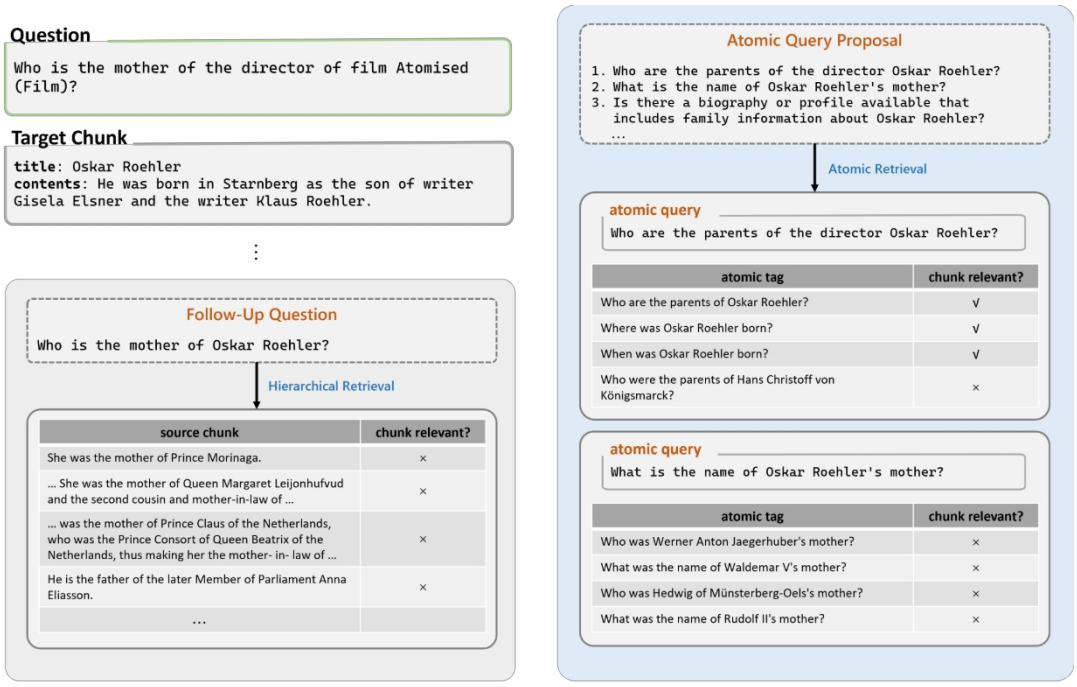


그림 2. Self-Ask와 KAR(3)의 사례 시연. KAR³은 여러 원자적 질의를 제안함으로써 관련 지식 청크를 효과적으로 검색하는 반면, Self-Ask가 사용하는 단일 결정론적 후속 질문 접근법은 지식 기반의 스키마와 일치하지 않아 검색 실패를 초래합니다.

이러한 문제를 완화하기 위해 RAG(Lewis et al., 2020)가 유망한 해결책으로 부상했으며, 생성된 콘텐츠를 고정시키기 위해 외부 지식 검색으로 대규모 언어 모델(LLM)을 보강합니다. RAG 프레임워크는 LLM 내에 인코딩된 지식을 보완하거나 대체함으로써 사실성과 관련성을 개선하는 것을 목표로 합니다. 그러나 기존 RAG 시스템은 종종 도메인 특화 복잡 작업에 어려움을 겪습니다. 예를 들어, "*HUMIRA의 최신 바이오시밀러 중 성공적으로 승인된 제품명을 제공해 주세요*"와 같은 질의에 답하려면 여러 출처에서 흩어진 전문 지식(예: *HUMIRA의 바이오시밀러*)을 검색하는 것 이상의 작업이 필요합니다. 이는 적격 제품과 승인 일정에 대한 논리적 추론을 통해 정확하고 신뢰할 수 있는 답변을 종합해야 합니다. 현재 RAG 방법은 주로 일반 텍스트 검색에 의존하는데, 이는 도메인 특화 표현 내 상관관계를 효과적으로 포착하지 못할 수 있으며, 종종 이용 가능한 지식을 고려하지 않은 채 질문 분해를 수행하여 최적화되지 않은 하위 질문 생성, 비효율적인 검색 및 추론 실패를 초래합니다.

본 연구에서는 RAG의 발전을 위해서는 지식 인식 처리, 특히 질문 분해 및 정보 검색에 대한 지식 인식 처리와 반복적 추론이 필요하며, 이를 통해 전문 분야에서 복잡한 다단계 질문을 효과적으로 해결할 수 있다고 주장한다.

도메인 특화 이해를 위한 지식 인식 처리 특화 도메인에서 복잡한 논리 기반 작업을 해결하려면 특정 도메인에서 복잡한 논리 기반 작업을 해결하려면 사용자의 정보 요구와 검색된 데이터의 근본적 맥락을 깊이 이해하기 위한 지식 추출 및 이해가 필요하다. 예를 들어 의학, 법률, 금융과 같은 분야의 전문적 질문은 종종 도메인 특화 용어와 논리를 포함하며, 일반적인 대규모 언어 모델(LLM)은 이를 완전히 파악하지 못할 수 있다. 키워드 매칭(Ram et al., 2023; Jiang et al., 2023)이나 임베딩 유사도(Gao et al., 2023)를 기반으로 텍스트 구절을 검색하는 기존 RAG 시스템은 문맥적으로 관련성 있는 정보를 검색할 수 있지만, 의미적 정확성이 부족하여 복잡한 질문에 답하기에는 불충분할 수 있습니다.

복잡한 질의 해결을 위한 반복적 추론 답변이 여러 출처의 정보를 종합하는 데 의존하는 복잡한 추론 작업은 원본 질문을 일련의 더 단순하고 상호 연관된 하위 질문들로 분해할 것을 요구한다(Press et al., 2023). 그럼에도 불구하고, 이 접근법은 지식 모델(LLM)이 지식을 쉽게 접근할 수 없는 영역에서는 장애물에 직면할 수 있다. 이러한 영역에서의 분해는 독립적 작업이 아닌 맥락적이어야 한다고 주장한다. 즉, 분해된 질의는 검색된 지식과 맥락을 점진적으로 활용해 답변할 수 있으며, 이후 질의를 정교화하는 방향으로 진화해야 한다. 이러한 반복적 접근은 시스템이 사용자의 질의에 대한 이해를 발전시켜 후속 질문이 가장 최근 검색 결과에 기반하도록 보장한다.

우리는 지식 인식 이중 재작성 및 추론 메커니즘을 활용하는 새로운 프레임워크인 KAR^3-RAG를 소개합니다. 본 접근법은 질문 재작성과 지식 검색 간의 동적 상호작용을 특징으로 하며, 그림 1에서 설명된 바와 같이 각 반복 단계에서 질의와 검색된 컨텍스트를 적응적으로 정교화할 수 있도록 합니다. 본 시스템의 핵심 구성 요소로는 원시 데이터를 더 세분화된 검색을 위한 원자 태그로 분해하는 지식 원자화기, 진화하는 컨텍스트를 기반으로 후속 질문을 생성하는 질의 제안기, 원자 지식 정렬을 기반으로 관련 지식을 식별 및 검색하는 원자 검색기, 검색된 정보를 바탕으로 가장 관련성 높은 후속 질문을 결정하는 원자 선택기가 포함됩니다. 특히, 원자 태그는 주어진 청크로 답변 가능한 관련 질의로 구성되어 청크의 다각적 지식을 포괄하고 효과적인 검색을 용이하게 합니다. 원자 질의 제안은 질문에 대한 답변을 개선하는 데 도움이 되는 지식을 탐색하기 위해 제기됩니다. 이러한 구성 요소를 활용함으로써 본 시스템은 질문과 검색된 지식에 대한 이해를 반복적으로 정교화하여 다중 단계에 걸쳐 더 정확하고 문맥을 고려한 추론을 가능하게 합니다. 그림 2에서 보여주는 실제 사례를 통해 작업 분해와 원자적 검색의 장점을 입증합니다. 본 접근법은 다각적인 작업 분해를 가능하게 할 뿐만 아니라, 원자적 태깅을 통해 코퍼스 구성과 질의 간의 불일치를 완화합니다.

우리의 주요 기여는 다음과 같습니다: 1) 검색된 지식을 질문 분해에 통합하여 추론 경로의 반복적 탐색을 가능케 하는 지식 인식형 RAG 프레임워크를 제안합니다. 2) 이중 작성을 통한 원자적 지식 정렬 접근법을 도입하여 질의 분해와 검색을 긴밀히 결합함으로써 검색 효율성을 크게 향상시킵니다. 3) 다양한 벤치마크 데이터셋에서 포괄적인 실험 및 제거 연구를 통해 본 접근법의 우수한 성능을 검증하였으며, 두 번째로 우수한 방법 대비 최대 20.4%의 성능 향상을 달성하였습니다.

2. 관련 연구

2.1. RAG

RAG는 외부 지식을 효과적으로 통합하여 LLM의 생성 성능을 향상시키는 유망한 솔루션으로 부상했습니다. 단순한 RAG 시스템은 외부 데이터 소스에서 관련 정보를 검색하여 질문 프롬프트의 맥락에 보충 지식으로 통합함으로써 문맥적으로 적절한 생성을 가능하게 합니다(Ram et al., 2023). 고급 RAG 접근법은 쿼리 최적화(Ma et al., 2023; Zheng et al., 2023), 다중 세분화 청크화(Chen et al., 2023; Zhong et al., 2024), 혼합 검색(Yang, 2023) 및 재정렬(Cohere, 2023)을 포함한다. 한편, 검색 성능 향상을 위해 명시적(Zheng et al., 2024) 또는 암시적(Gao et al., 2022)으로 쿼리 재작성에 주력하는 연구도 있다. 다른 한편으로는, 여러 연구에서 원시 데이터 소스를 구조화된 데이터로 변환하여 궁극적으로 더 효과적인 검색 및 추론을 위한 가치 있는 지식으로 전환하고 있다(Wang et al., 2023b; Zheng et al., 2024; Raina & Gales, 2024; Liang et al., 2024). 본 시스템에서는 쿼리와 청크 모두에 대해 원자적 재작성(atomic rewriting)을 도입하여 다중 세분성 질문 분해를 달성하고 청크에서 내재된 지식을 포괄적으로 추출합니다.

요약(Hayashi et al., 2021) 및 다중 단계 추론(Ho et al., 2020)과 같은 복잡한 작업을 해결하기 위해 최근 연구는 기존 RAG 모듈을 활용하여 이러한 과제를 공동으로 해결하는 고급 조정 체계 개발에 초점을 맞추고 있다. Iter-RetGen (Shao et al., 2023)과 DSP (Khattab et al., 2023)는 검색-읽기 반복을 사용하여 생성 응답을 다음 라운드 검색의 컨텍스트로 활용합니다. FLARE (Jiang et al., 2023)는 신뢰도 기반 능동 검색 메커니즘을 제안합니다. 본 연구의 접근법은 컨텍스트 인식 추론 프로세스를 활용하는 반복 기반 파이프라인을 채택하여, 각 반복에 대한 후속 질문을 적응적으로 구성하고 복잡한 작업의 검색 및 추론 난이도를 낮춥니다.

2.2. 다중 단계 질의응답

다중 정보 기반 질문 답변(MHQA)(Yang et al., 2018)은 종종 서로 다른 출처에 흩어져 있는 여러 정보 조각에 대한 추론을 요구합니다. 이 작업은 관련 정보를 검색할 뿐만 아니라 검색된 조각들을 효과적으로 결합하고 추론하여 올바른 답에 도달해야 하므로 독특한 도전 과제를 제시합니다. 기존 MHQA의 그래프 기반 접근법은 그래프를 구축하고 그래프 신경망(GNN)을 통해 추론하여 답변을 예측하는 방식으로 문제를 해결해왔다(Qiu 외, 2019; Fang 외, 2020). LLM의 등장과 함께 최근 그래프 기반 방법(Li & Du, 2023; Panda et al., 2024; Liang et al., 2024)은 검색을 위한 KG 구축과 LLM을 통한 응답 생성을 위해 진화했습니다. 그러나 고품질의 도메인 특화 KG 구축은 비용이 많이 들고, 구조화된 트리플 형식은 문맥 표현에 본질적인 제약을 가해 표현력을 제한합니다. Self-RAG(Zhang et al., 2024a)와 빙 검색(Asai et al., 2023)은 MHQA를 감독형 문제로 다루어 라벨링된 데이터와 추가 훈련을 필요로 합니다.

또 다른 방법론 계열은 다중 경로 질문을 체인형 경로(Trivedi et al., 2023; Khattab et al., 2023; Feng et al., 2023; Xu et al., 2024) 또는 트리형 경로(Zhang et al., 2024b; Jiapeng et al., 2024; Cao et al., 2023)를 따라 하위 질문으로 분해합니다(그림 1 참조). 하위 질문들은 순차적 청크 재구성(chunk re-

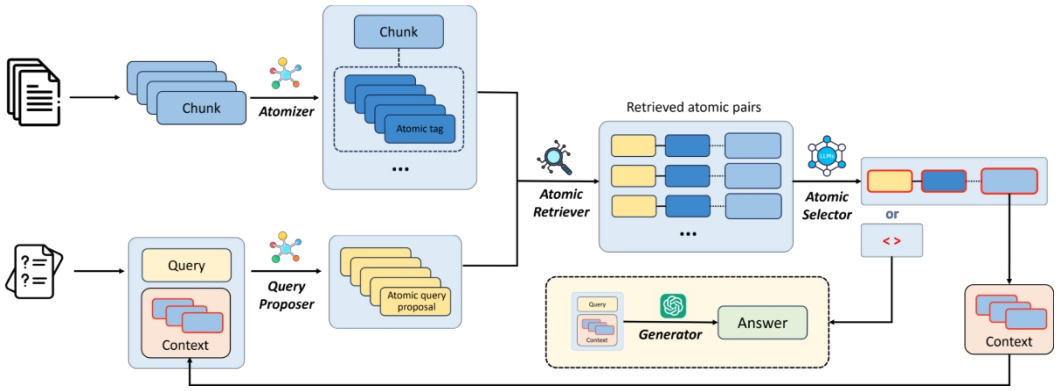


그림 3. KAR³-RAG 워크플로우 개요. 원자화기를 통한 지식 원자화, 쿼리 제안기를 활용한 지식 인식 작업 분해, 원자 검색 및 원자 선택기를 보여줍니다. 질의 제안기는 원본 질문과 참조 문맥을 기반으로 원자적 질의 제안을 생성한다. 이 제안들은 관련 원자적 태그를 검색하는 데 사용되어 검색된 원자적 쌍을 생성한다. 원자 선택기는 가장 관련성 높은 쌍과 해당 청크를 선택하며, 이는 후속 반복에서 작업 분해를 위해 참조 문맥에 추가된다. 원자 선택기가 추가 정보가 필요하지 않다고 판단하고 원자적 쌍이 선택되지 않으면, 원본 질문과 참조 문맥이 생성기로 전달되어 최종 답변을 생성한다.

검색된 결과가 이후 추론 과정을 촉진하는 검색 방식이다. 체인형 분해에서는 단일 하위 질문이 생성되며, 그 답변 가용성이 보장되지 않아 답변 실패로 이어질 수 있다. 반면 트리형 분해는 다중 추론 경로 탐색을 요구하며, 최종 응답 생성을 위해 정교한 증거 검증 및 융합이 필요하다. 본 접근법은 원자 검색의 관련성을 기반으로 제안된 질의 집합에서 하위 질문을 상호작용적으로 선택함으로써 추론 경로를 탐색한다. 이는 업데이트된 맥락을 활용하고 이용 가능한 지식을 가진 질의 제안을 선택함으로써 유연한 분해를 가능하게 한다.

3. 방법론

3.1. 예비

RAG 시스템에서 텍스트 코퍼스는 문서 청크 집합으로 분할되며, 이를 $D = \{d_1, d_2, ..., d_n\}$로 표기한다. 여기서 d_i는 i번째 문서 청크를 나타낸다. 원본
질문은 q로 표시되며, 이에 대응하는 정답은 a로 표현된다. 검색 단계에서는 질문 q와 각 문서 조각 d_i간의 유사도를 평가한 후, 상위 k개 관련성 높은 조각을 검색 결과로 선정하여 후속 생성 작업의 기초를 마련한다.

$$\mathrm{R} : \underset{\mathrm{di}\in D}{\text{topk}}\ \mathrm{Sim}(q, d_i) \rightarrow D^q \qquad (1)$$

여기서 검색기 R은 유사도 함수 $Sim(\cdot)$을 기반으로 상위 k개 관련성 높은 청크 D^q를 선택합니다. 마지막으로 원본 질문과 검색된 청크는
대규모 언어 모델에 입력되어 답변을 생성하며, 이를 다음과 같이 표기합니다.

$\hat{a} = \mathrm{LLM}(q, D^q)$. 고급 RAG 시스템에서는 쿼리 재작성을 통해 질문과 검색 대상 챕터 간의 의미적 간극을 해소합니다. 재작성된 쿼리는 $\hat{q} = f_{re}(q)$로 표현됩니다. 고급 RAG 시스템의 워크플로는 다음과 같이 추가 개선됩니다.

$$\hat{a} = \mathrm{LLM}(q, D^{\hat{q}}), \text{ 여기서 } D^{\hat{q}} = \mathrm{R}(\hat{q}, D) \qquad (2)$$

이러한 개선을 통해 시스템은 쿼리를 관련 문서 청크와 더 잘 정렬하여 검색 정확도와 답변 생성을 향상시킬 수 있습니다. 그러나 복잡한 다중 단계 질문을 처리하는 것은 여전히 어려운 과제입니다. 이러한 질문은 종종 여러 청크에 걸친 추론과 여러 검색 및 생성 단계를 통한 정보 통합을 요구하는데, 이는 단일 패스로는 완전히 포착하기 어려울 수 있습니다.

3.2. 프레임워크

복잡한 다중 단계 질문을 해결하기 위해, 지식 인식 **이중** 재작성 및 추론(Knowledge-Aware **dual** Rewriting and Reasoning)을 적용한 향상된 RAG 시스템인 KAR(3)을 제안한다. 이 시스템은 반복적인 검색-추론-생성 메커니즘을 활용하여 관련 정보를 점진적으로 수집하고 증분적 컨텍스트에 대한 점진적 추론을 가능하게 한다. 제안된 워크플로의 개요는 그림 3에 제시되어 있습니다. 본 프레임워크에서는 원시 데이터 청크를 지식 원자화기를 통해 원자 태그로 분해하여 후속 검색을 위한 원자 지식 기반을 구축합니다. 질문 역시 쿼리 제안기를 통해 원자화되어 원자 쿼리 제안들을 생성하며, 이는 지식베이스에서 관련 원자 태그를 검색하는 데 활용됩니다. 의미적 격차를 해소하고 지식 정렬을 개선하기 위해 덩어리와 질문 모두 재작성됩니다. 이후 원자 검색기가 각 원자 쿼리 제안에 대해 *상위 k개의* 원자 쌍을 선별합니다. 검색된 원자 쌍을 기반으로 추론기 역할을 하는 원자 선택기가 가장 유용한 원자 쌍을 식별합니다.

문제 해결을 위해 해당 원시 청크를 컨텍스트에 추가합니다. 이 컨텍스트는 다음 반복에서 작업 분해를 위해 원본 질문과 통합됩니다. 반복 과정은 저품질 질의 제안 생성이나 관련 원자 태그 후보 부족으로 적합한 원자 태그를 검색하지 못할 경우 조기에 종료될 수 있습니다. 이때 원본 질문과 컨텍스트는 생성기로 전달되어 최종 답변을 생성합니다.

3.3. 지식 원자화

분할된 텍스트는 종종 다각적인 정보를 포함하며,

일반적으로 특정 작업을 처리하기 위해서는 하위 집합만 필요합니다. 통합된 전통적인 정보 검색 방법들은
단일 청크 내의 모든 정보를 날짜로 표시하는 것은
필요한 정확한 정보를 효율적으로 검색하는 데 도움이 되지 않을 수 있습니다.
최근 연구에서는 트리를 지식 추출을 탐구해 왔습니다.

챕터화된 텍스트로부터 에지 유닛을 추출하고 효율적인 정보 검색을 용이하게 하기 위해 지식 그래프를 구축하는 방법(Edge et al., 2024; Panda et al., 2024). 그러나 이러한 지식 그래프 구축은 비용이 많이 들며, 내재된 지식이 항상 완전히 탐색되지는 않을 수 있다. 문서에 내재된 지식을 보다 효과적으로 제시하기 위해, 우리는 지식 추출을 위한 원본 문서의 원자화(atomizing)를 제안하며, 이를 *지식 원자화*(*Knowledge Atomizing*)라고 명명한다. 이 접근법은 LLM의 문맥 이해 및 콘텐츠 생성 능력을 활용하여 각 문서 청크 내의 원자적 지식 조각을 자동으로 태깅한다.

원자적 지식의 표현 방식은 다양할 수 있다. 선언적 문장이나 주어-관계-목적어 튜플을 활용하는 대신, 우리는 질문을 지식 인덱스로 사용하여 저장된 지식과 질의 간의 간극을 더욱 좁힐 것을 제안한다. 지식 원자화 과정에서 문서 청크를 LLM에 컨텍스트로 입력하고, 주어진 청크로 답할 수 있는 관련 질문을 최대한 많이 생성하도록 요청합니다. 이렇게 생성된 원자 태그는 주어진 청크와 함께 저장됩니다. 지식 원자화기는 각 청크에 원자화 연산을 적용합니다.

$$f_a(d_k) = \{q_{k1}, q_{k2}, \cdots, q_{km}\} \tag{3}$$

원자 태그는 원자화기에 의해 모든 청크에 대해 생성되어 원자 지식 기반을 형성하며, 이는 KB = 로 표시됩니다.
$\{f_a(d_k), d_k\}$. 지식 원자화의 예는 그림 4(a)에
원자 태그는 덩어리 내에 포함된 지식의
각 덩어리에는 여러 원자 태그가 부여되므로, 원자 쿼리를 통해 관련 원자 태그를 찾아내고, 이를 통해 연관된 참조 덩어리로 연결될 수 있다.

3.4. 지식 기반 작업 분해

복잡한 다중 단계 질문을 해결하려면 종종 여러 지식 조각을 통합해야 하며, 이는 암묵적으로

알고리즘 1 지식 인식 분해를 통한 과제 해결

1: 컨텍스트 초기화 $C_0 \leftarrow \phi$
2: **for** $t = 1, 2, \ldots, N$ **do**
3: 원자적 질의 제안 생성 $\mathbf{q}^t \leftarrow f_p(q, C_{t-1})$
4: 각 원자 쿼리에 대해 *상위 k개의* 원자 쌍을 검색
지식베이스로부터 제안

$$P \leftarrow R_{원자}(\mathbf{q}^t, f_a^t(D))$$

5: 추가 정보가 불필요한 경우 가장 유용한 원자 태그 또는 *None* 선택

$$q_{ksls} \leftarrow \text{LLM}(q, C_{t-1}, P)$$

6: q_{kl} *None*이면
7: $C_t \leftarrow C_{t-1}$
8: **분해**
9: **else**
10: q_k에 해당하는 관련 청크 c^t를 가져옵니다.
11: 컨텍스트 C_t업데이트 $\leftarrow C_{t-1} \cup c^t$
12: **end if**
13: **for 종료**
14: 답변 생성 $a \leftarrow \text{LLM}(q, C_t)$

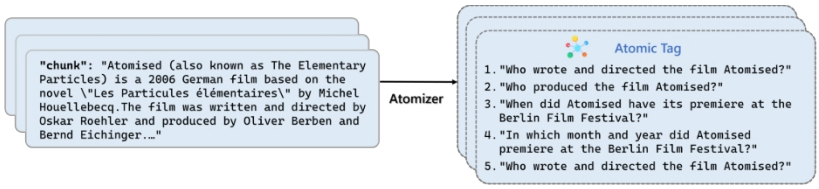
원본 질문을 검색을 위한 여러 순차적 또는 병렬적 원자 태그로 분해하는 능력이 요구됩니다. 우리는 이 작업을 *작업 분해*(Task *Decomposition*)라고 부릅니다. 추출된 원자 지식과 원본 청크를 결합하여 원자 지식 기반을 구축합니다. 작업이 분해될 때마다 원자 지식 기반은 이용 가능한 지식에 대한 통찰력을 제공하여 지식 인식 작업 분해를 가능하게 합니다. 우리는 *지식 인식 작업 분해* 워크플로를 설계하며, 작업 해결을 위한 완전한 알고리즘은 알고리즘 1에 상세히 기술되어 있고, 예시는 그림 4(b)에 설명되어 있습니다.

초기 참조 컨텍스트 $C_{(0)}$은 빈 집합으로 초기화된다. 첫 번째 반복에서 작업 분해는 원본 질문만을 기반으로
원본 질문만을 기반으로 원자적 질의 제안을 생성합니다. 반복이 진행됨에 따라 t번째 반복 시점에 누적된 컨텍스트
C_{t-1} 으로 표기되며, 이전 반복에서 검색된 청크들로 구성됩니다.
추출된 청크들로 구성됩니다. t번째 반복 동안, 쿼리 제안기는
원본 질문과 누적된 컨텍스트를 기반으로 원자적 질의 제안을 생성한다.

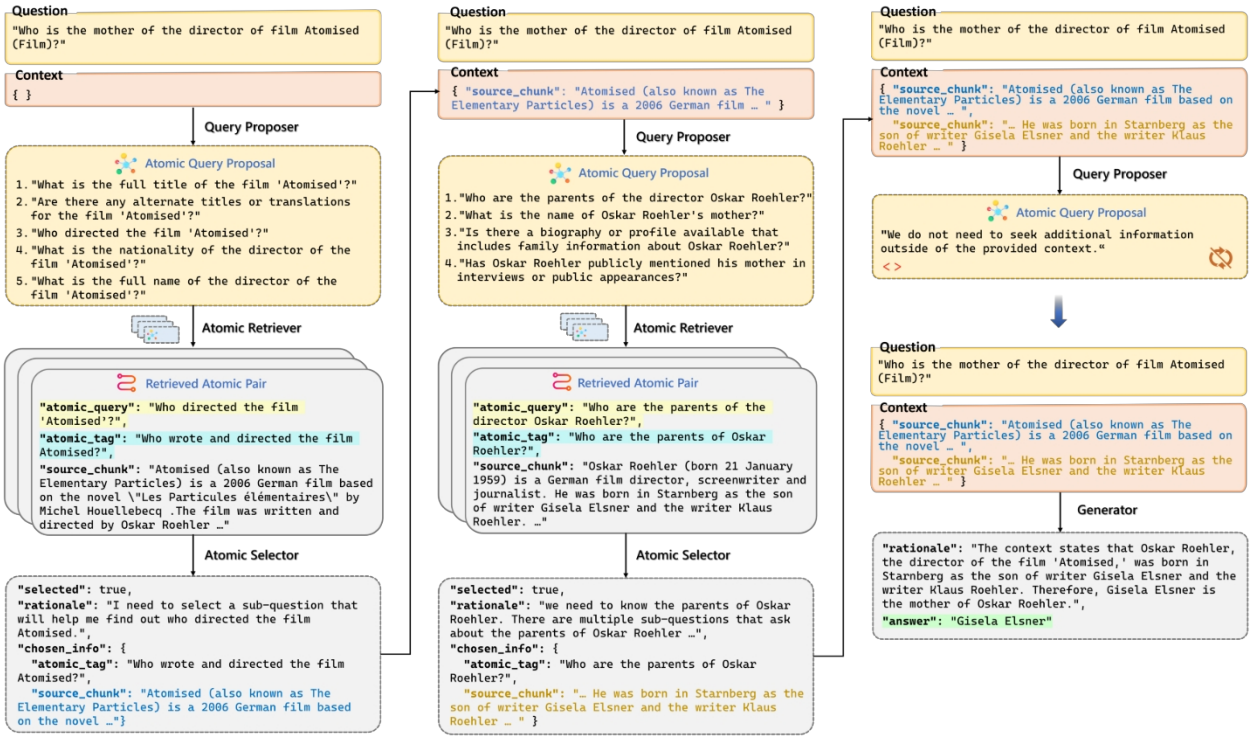
$$f_p(q, C_{t-1}) = \{q^t_1, q^t_2, \cdots, q^t_n\} \tag{4}$$

쿼리 제안기 $f_p(\cdot)$는 LLM 또는 학습 가능한 구성 요소로 구현될 수 있다. 우리는 LLM을 활용하여 작업 완료에 잠재적으로 유익한 쿼리 제안(q^t =$\{q^t\}$)을 생성한다. 이 과정에서 선택된 참조 청크 C(t) (−) (1) 이 제공된다.

작업 완료에 잠재적으로 유익한 쿼리 제안($\mathbf{q}^t$ =$\{q^t\}$)을 생성합니다. 이 과정에서 선택된 참조 청크 C_{t-1} 은 컨텍스트로 제공됩니다.



(a) Illustrative example of of knowledge atomizing



(b) Illustrative example of KAR^3-RAG case

그림 4. KAR³-RAG 사례의 예시: (a) 지식 원자화 사례, (b) 지식 인식 작업 분해를 통한 RAG 사례. 반복 과정이 진행됨에 따라, 원자적 검색 및 선택을 통해 관련 정보 조각을 축적함으로써 참조 컨텍스트가 풍부해진다. 컨텍스트가 확장됨에 따라 생성되는 원자적 쿼리 제안의 수는 더 이상 제안이 생성되지 않을 때까지 감소한다. 이후 반복 과정이 종료되고, 결합된 질문과 컨텍스트를 활용하여 최종 응답을 생성한다.

이미 알려진 지식과 연결된 제안 생성을 피하기 위한 텍스트. 결과적으로 쿼리 제안은 반복마다 진화하며, 업데이트된 컨텍스트에 적응하고 컨텍스트 내 청크를 넘어 추가 지식을 탐색하는 것을 목표로 합니다. 각 원자적 쿼리 제안에 대해, 지식베이스에서 *상위 k개* 관련 원자적 태그 후보와 그 출처 청크를 검색합니다. 원자적 검색 과정은 다음과 같습니다:

$$\mathrm{R}_{atom}: \underset{q_{kl} \in \mathcal{T}_a(\mathrm{D})}{\mathrm{topk}} \mathrm{Sim}(\hat{q}_i^t, q_{kl}) \longrightarrow P^{\hat{q}_i^t}_{\mathrm{KB}} \quad (5)$$

원자 검색기(R_{atom}으로 표기)는 각 원자 쿼리 제안에 대해 검색된 원자 쌍 집합을 생성하며, 이는 $P^{\hat{q}_i^t} = \{(\hat{q}_i^t, q_{kl}, d_k)\}$로 표현된다. 모든 검색된 원자 쌍으로 선택된 원자 쌍의 관련 원시 청크를 추가로 검색합니다. 각 원자 쿼리 제안에서 추출된 이 청크들은 t번째 반복에서 추가된 컨텍스트 c_t로 표시됩니다.

해당 임베딩 간의 유사도를 기반으로 *상위 k개* 원자 태그를 검색하며, 제안된 원자 태그와의 유사도가 지정된 임계값 δ 이상일 경우에만 포함됩니다. 원본 질문, 누적된 컨텍스트, 검색된 원자 쌍 목록을 바탕으로 원자 선택기는 LLM을 활용하여 문제 해결에 가장 유용한 원자 쌍을 선별합니다.

$$\mathrm{LLM}(q, C_{t-1}, P^{\hat{q}_i^t}) = (\hat{q}_i^t, g_{ks_{ts}}, d_{ks}) \quad (6)$$

원자 선택기(S_{atom}으로 표시)는 새로 제안된 원자 쌍에서 선택된 원자 쌍의 관련 원시 청크를 추가로 검색합니다.

게이트 처리하여 전체 집합 $P^{\hat{q}^{(t)}}$를 생성한다. 코사인 chunk는 d에 대응합니다 $_{ks}$에 해당하는 청크를 추출합니다.

추출 과정은 다음 공식으로 표현될 수 있다.

$$c_t = \mathrm{S}_{atom}\left(\mathrm{R}_{atom}\left(f_p\left(q, \mathrm{C}_{t-1}\right), f_a\left(\mathrm{D}\right)\right)\right) \quad (7)$$

이렇게 검색된 청크는 다음 분해 단계의 참조 컨텍스트로 통합되며, 다음과 같이 표현됩니다.

$\mathrm{C}_t = c_t \cup \mathrm{C}_{t-1}$. 지식 인식 분해는 이를 통해
최대 N회까지 반복할 수 있으며, 여기서 N은 계산 비용을 제어하기 위해 설정된 하이퍼파라미터입니다.
반복 과정은 저품질 쿼리 제안 생성이나

관련 원자 태그 후보가 존재하지 않아 적합한 원자 태그를 검색하지 못할 경우, 또는 LLM이 누적된 지식이 작업 완료에 충분하다고 판단할 경우 프로세스가 중단될 수 있습니다. 이러한 조기 종료 메커니즘은 모든 반복을 완료하기 전에 프로세스를 종료할 수 있게 하여 계산 비용을 절감합니다.

종료 메커니즘은 모든 반복을 완료하기 전에 프로세스를 종료할 수 있게 하여 계산 비용을 절감합니다.

정확도를 저하시키지 않으면서. 마지막으로, 누적된 컨텍스트 C_t는 1행에서 주어진 질문 q에 대한 답변 a^를 생성하는 데 활용된다.

지식 인식 분해는 학습 가능한 구성 요소일 수 있다는 점을 언급할 가치가 있습니다. 각 전문 지식 기반에 대해, 우리는 각 분해 단계에서 수집된 데이터를 활용할 수 있습니다.

위치 반복—구체적으로는 $(q, a, \hat{a}, \{q_s^t, c^t, \tilde{q}^t, P^{q^t}, \mathrm{C}_t\})$.
이렇게 훈련된 제안자는 추론 과정에서 원자 쿼리 q'를 직접 제안할 수 있습니다. 이는 알고리즘 1의 1~1행이 이 학습된 제안자에 대한 단일 호출로 대체될 수 있음을 의미하며, 이를 통해 추론 시간과 계산 비용을 모두 줄일 수 있습니다. 효율적인 쿼리 제안자 훈련에 대한 탐구는 향후 연구 과제로 남겨둡니다.

4. 평가 및 지표

KAR³은 특수화된 도메인의 과제를 해결하기 위해 제안되었으므로, 우리는 중국 법률 벤치마크인 LawBench와 호주 공개 법률 QA 벤치마크 모두에서 실험을 수행했습니다. 실험 결과 KAR³은 모든 벤치마크에서 기존 방법 대비 상당한 성능 향상을 보였으며, 생성 작업 정확도는 LawBench에서 90.12%, 호주 법률 QA에서 98.59%에 달했습니다. 해당 법률 벤치마크 및 실험 결과에 대한 상세 설명은 부록 A.8에서 확인할 수 있습니다.

제안된 접근법을 기준 방법과 더 잘 비교하기 위해, 본 절에서는 널리 인정받는 오픈 도메인 벤치마크에 집중한다. 4.1절과 4.2절은 각각 실험 설정과 주요 실험 결과를 개괄한다. 제거 연구는 4.3절에서 논의된다. 또한 내용 제약으로 인해 비용 분석과 사례 연구는 부록 A.5와 A.6에 포함되었다.

4.1. 실험 설정

방법론 제안된 지식 인식 분해 접근법의 성능을 철저히 평가하기 위해, LLM을 활용한 과제 해결을 위한 다양한 전략을 대표하는 여러 기준 방법을 선정하였다. 추가적인 컨텍스트 없이 기본 LLM의 내재적 추론 능력과 내장 지식을 평가하기 위해 **제로샷 CoT**(Kojima et al., 2022)를 포함하였다. 검색을 통해 외부 지식을 도입하는 **Naive RAG**(Lewis et al., 2020)는 증강된 지식의 점진적 이점을 평가하기 위한 벤치마크 역할을 합니다. **Self-Ask** 프레임워크(Press et al., 2023)는 반복적 질문 분해 및 답변 전략이 작업 수행에 미치는 영향을 조사하기 위해 활용됩니다. 다단계 질문 처리를 위한 근거를 반복적으로 생성하는 **IRCoT**(Trivedi et al., 2023)와 최근 응답을 반복적으로 검색 쿼리로 활용하여 응답 품질을 개선하는 **Iter-RetGen** (Shao et al., 2023)은 최근 응답을 검색 쿼리로 반복 활용하여 응답 품질을 개선하며, **ProbTree**(Cao et al., 2023)는 복잡한 QA를 검색 트리로 명시적으로 분해합니다. 이들 역시 성능 비교를 위해 수행되었습니다. 방법론에 대한 상세한 설명은 부록 A.4에 제시됩니다.

본 실험에서는 GPT-4(1106-Preview)와

Llama-3.1-70B-Instruct를 사용했습니다.
. 제4.2절에 제시된 실험을 위해,

반복 횟수 N은 Self-Ask with Retrieval, IRCoT, Iter-RetGen 및 KAR³에 대해 5로 설정됩니다. 또한 원자 검색기는 $k = 4$ 및 $\delta = 0.5$로 초기화됩니다. 검색 및 LLM에 대한 하이퍼파라미터의 포괄적인 목록은 부록 A.3에서 확인할 수 있습니다. 간결함을 위해 Llama-3.1-70B-Instruct는 이후 내용에서 Llama 3으로 약칭됩니다.

지표 기존 벤치마크와의 일관성을 보장하기 위해, 본 실험 평가에서는 **F1** 점수를 표준 지표로 채택합니다. 단순 어휘 일치 이상의 수준에서 응답과 의도된 정답 간의 정합성을 보다 정확히 평가하기 위해, *GPT-4*를 활용한 새로운 평가 지표를 도입합니다. 이 과정에서 *GPT-4*는 평가자 역할을 수행하며, 질문과 정답 라벨에 대한 응답의 정확성을 평가합니다. 이 지표를 **정확도(Acc)**라고 명명합니다. 샘플 세트에 대한 수동 검증을 통해 *GPT-4*의 판단이 인간 평가자와 완전히 일치함을 확인하여 이 지표의 신뢰성을 입증했습니다. 또한 부록 A.4에서 완전 일치(EM), 재현율(Recall), 정밀도(Precision)를 포함한 전체 평가 결과를 확인할 수 있습니다.

데이터셋 기준 방법과의 비교를 위해 본 연구의 평가는 널리 인정받는 세 가지 다중 홉 데이터셋에 초점을 맞췄습니다: HotpotQA (Yang et al., 2018), 2WikiMulti-HopQA (Ho et al., 2020), 그리고 MuSiQue (Trivedi

et al., 2022). 이 데이터셋들에 대한 간략한 소개는 부록 A.1에서 확인
할 수 있습니다. 각 데이터셋에 대해 무작위로 샘플링합니다.

표 1. GPT-4를 활용한 다중 홉 QA 데이터셋 성능 비교. 최고 성능은 굵게, 차순위는 밑줄 처리.

표 2. Llama 3 기반 다중 홉 QA 데이터셋 성능 비교. 최고 성능은 굵게, 차순위는 밑줄 표시.

개발 세트에서 500개의 QA 데이터를 무작위로 추출하며, 무작위성을 보장하기 위해 질문 유형과 홉 수를 고려하지 않습니다. 추출된 모든 QA 데이터의 컨텍스트 문단을 각 벤치마크별로 단일 지식 기반으로 통합하여 더 복잡한 검색 시나리오를 생성합니다. 이 설계 선택은 모델의 작업 분해 및 관련 컨텍스트 검색 능력을 엄격하게 평가하기 위한 것입니다. 간결성을 위해 2WikiMultiHopQA는 2Wiki로 약칭합니다.

4.2. 주요 결과

표 1과 표 2(1)에서 확인할 수 있듯이, 본 접근법은 GPT-4와 Llama 3 모두에서 모든 데이터셋에 걸쳐 우수한 성능을 달성합니다. 특히 GPT-4의 경우 약 +1.4(1.6%), +2.2(2.8%), 그리고
HotpotQA, 2Wiki, MuSiQue의 두 번째로 우수한 결과 대비 정확도에서 각각 +7.0(12.6%), +3.4(4.4%), +10.1(20.4%)의 향상을 달성했습니다. 마찬가지로 Llama 3을 사용했을 때도 각각 +0.2(0.2%), +3.4(4.4%), +10.1(20.4%)의 증가를 보였습니다. 이러한 향상은 통계적으로 유의하며, KAR(3)이 복잡한 QA 작업을 처리하는 데 있어 견고함을 입증합니다.
+10.1(20.4%)의 정확도 향상을 각각 달성했습니다. 이러한 향상은 통계적으로 유의하며, KAR³이 복잡한 QA 작업을 처리하는 데 있어 견고함을 강조합니다.

제안된 접근법인 KAR³은 지식 기반 작업 분해를 강조하며, Self-Ask에서 사용된 주어진 데모에 의존하는 자발적 분해 메커니즘과 차별화됩니다. 이는 이용 가능한 지식을 인지하며 분해를 수행하고, 원자 태그를 중간 매개체로 효과적으로 활용하여

[1] Llama 3 엔드포인트에서 로그 확률(logprobs)을 얻는 데 문제가 발생했기 때문에, Llama 3을 사용한 ProbTree 실험은 향후 연구 과제로 남겨둡니다.

의미론적 간극. 알고리즘 1에 상세히 기술된 "제안 후 선택" 프레임워크는 동적 분해 경로 탐색을 가능하게 하며, 질문의 의도를 검증하고 과거 근거 생성 과정의 잠재적 오류를 수정할 기회를 제공한다. 이 점의 실제 적용 사례는 부록 A.6의 사례(a)에서 확인할 수 있습니다. 결과적으로 실험 결과는 KAR³이 다양한 모델을 사용하는 다른 방법들을 지속적으로 능가함을 보여주며, 이는 복잡한 추론 시나리오에서 그 효과성뿐만 아니라 다양한 모델에 대한 견고성과 적응성을 입증합니다.

방법	HotpotQA		2Wiki		MuSiQue	
	F1	정확도	F1	Acc	F1	Acc
제로샷 CoT	43.94	53.60	41.40	43.87	22.90	23.47
순진한 RAG	72.67	82.60	59.74	62.80	43.31	44.40
R.와 함께하는 자기 질문	[illegible]	80.00	[illegible]	75.00	[illegible]	51.40
IRCoT	67.30	81.00	63.83	70.40	47.57	49.20
Iter-RetGen	75.27	86.60	67.21	73.60	52.48	55.60
ProbTree	62.41	73.40	69.42	80.00	43.26	52.86
KAR³ (우리)	**76.48**	**88.00**	**75.00**	**82.20**	**57.86**	**62.60**

4.3. 제거 연구

***N*의 선택.** 우리는 먼저 반복 상한 *N을* 1, 2, ..., 10으로 설정하여 실험을 수행했으며, 그 결과는 그림 5에 제시되어 있습니다. 상세한 성능 지표는 부록 A.4의 표 8에서 확인할 수 있습니다. 세 데이터셋 모두에서 지원 사실 회상률과 답변 정확도 모두 일관된 상승 추세를 보입니다. 이러한 패턴은 특히 문제 해결에 더 상세하고 문맥적으로 관련성 높은 정보가 필요할 때, 추가 반복을 통해 출력을 점진적으로 향상시키는 본 접근법의 능력을 강조한다.

방법	HotpotQA		2Wiki		MuSiQue	
	F1	정확도	F1	Acc	F1	Acc
제로샷 CoT	40.10	54.80	38.54	43.20	15.69	19.80
순진한 RAG	70.78	84.20	56.58	62.20	32.53	36.40
자기 질문 w/ R.	70.25	83.00	66.25	74.00	38.19	44.20
IRCoT	[illegible]	88.00	[illegible]	77.60	[illegible]	49.60
Iter-RetGen	72.23	85.20	59.21	65.00	37.16	40.40
KAR³ (우리)	**75.27**	**88.20**	**72.79**	**81.00**	**50.68**	**59.70**

또한, 반복 횟수와 지원 사실 회상률의 관찰된 증가 사이의 관계를 검토한 결과, HotPotQA 및 2Wiki 데이터셋의 경우 회상 곡선이 네 번째 반복까지 현저한 증가를 보인다는 점을 확인했습니다. 반면 MuSiQue 데이터셋의 회상률은 부록 A.1에서 언급된 바와 같이 질문당 최대 홉 수가 4로 제한되어 있음에도 불구하고, 이 지점을 넘어선 후에도 계속해서 급격히 상승합니다. 이러한 차이는 KAR³이 제한된 반복 횟수 내에서 관련성 있고 유용한 정보를 검색하는 데 능숙하지만 여전히 한계가 있음을 시사합니다: KAR³은 사용된 대규모 언어 모델(LLM)의 추론 능력에 의존하며, 특히 질문의 복잡성이 증가함에 따라 필요한 정보를 완전히 포착하기 위해 추가 반복이 필요할 수 있습니다.

알고리즘 1은 조기 종료 메커니즘을 포함하고 있지만, *N 값이* 높아질수록 계산 요구량이 증가하는 것은 피할 수 없다. 따라서 4.2절의 실험에서는 계산 자원과 성능 향상 기대치 사이의 섬세한 균형을 이루기 위해 최대 홉 수보다 약간 높은 값인 $N = 5$를 선택하였다.

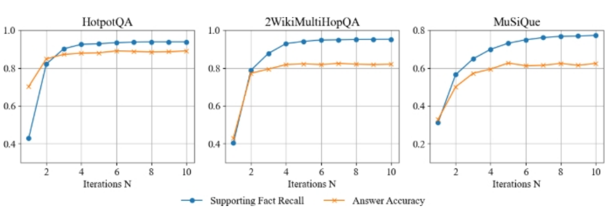


그림 5. 반복 횟수에 따른 보조 사실 회상률(파란색)과 답변 정확도(주황색).

표 3. KAR³구성 요소에 대한 제거 연구.

변수 구성 요소	수정	HotpotQA		2Wiki		MuSiQue	
		F1	정확도	F1	Acc	F1	Acc
지식 분무기	질문으로 분해 → 일반 텍스트로 분해	73.05	84.50	64.18	69.80	50.72	55.20
쿼리 제안자	여러 쿼리 제안 → 단일 쿼리 제안	75.06	85.60	70.19	76.40	49.67	52.20
원자적 리트리버	(원자 태그, 청크) 쌍 검색 → 청크 검색	76.31	86.60	67.14	72.40	49.05	53.00
원자 선택기	원자 태그로 청크 선택 → 청크 직접 선택	72.80	83.20	61.65	65.80	49.31	53.40
KAR³ (우리)		76.48	88.00	75.00	82.20	57.86	62.60

접근 방식 구성 요소의 기여도. KAR³은 네 가지 핵심 구성 요소로 이루어져 있습니다: 지식 분해기, 질의 제안기, 원자 검색기, 원자 선택기. 우리는 이러한 구성 요소들의 개별적 및 집단적 기여도를 확인하기 위해 제거 실험을 수행합니다.

이러한 구성 요소를 하나씩 수정하여 여러 방법 변형을 도입함으로써: (1) 지식 원자화기(knowledge at-omizer)의 경우, 원자적 태그 표현을 원자적 질문에서 일반 텍스트 문장으로 변경하여 원자적 지식 표현의 영향을 탐구합니다; (2) 질의 제안기(query pro-poser)의 경우, 원래 설계된 다중 제안 메커니즘의 장점을 평가하기 위해 단일 질의만 생성하도록 제한합니다; (3) 원자 검색기(atomic retriever)의 경우, (원자 태그, 청크) 쌍 대신 청크 단위로 검색하도록 구성 요소를 수정했습니다; (4) 원자 선택기(atomic selector)의 경우, 원자 태그로 청크를 필터링하는 대신 청크를 직접 선택하는 변형을 구현했습니다. 이 설정에서는 원자 태그가 존재하지 않으므로, 이후 컨텍스트 선택은 청크 자체에 의해 결정됩니다.

표 3의 결과에서 알 수 있듯이, 구성 요소들의 개별 기여도를 평가하였다. 지식 분해기, 질의 제안기, 원자 검색기 및 원자 선택기를 대체 구성 요소로 교체할 경우 세 데이터셋에서 각각 최대 15.1%, 16.6%, 15.3%, 16.2%의 정확도 저하가 발생함을 관찰했습니다. 이러한 제거 연구는 최적의 검색 성능과 일관된 추론 경로를 달성하기 위해 설계된 각 구성 요소가 필수적임을 시사합니다.

한계 논의. 복잡한 질문에 대한 핵심 정보 추출을 위한 추가 반복의 필요성 외에도, 부록 A.4의 표 9에 상세히 기술된 GPT-3.5 실험은 대규모 언어 모델(LLM)의 추론 능력에 의존하는 데 한계가 있음을 시사합니다. GPT-3.5를 사용한 KAR³의 성능은 IRCoT 및 Self-Ask w/ Retrieval과 같은 방법들을 크게 능가하지 못하며, 때로는 Self-Ask w/ Retrieval에 비해 부족하기도 합니다. 이는 KAR³의 성공이 고급 추론 기술과 복잡한 지시를 견고하게 따르는 능력에 달려 있음을 강조합니다.

오픈소스 모델 Llama 3을 사용한 실험 결과는 주목할 만한 성능 향상을 보여줍니다. 기준 방법들에 비해, 부록 A.5의 표 10에 상세히 기술된 바와 같이 본 접근법은 평가된 일부 방법들에 비해 더 높은 토큰 소비량을 요구합니다. 구체적으로 MuSiQue 데이터셋에서는 ProbTree 및 IR-CoT보다 적은 토큰을 사용하지만, 검색을 동반한 Iter-RetGen 및 Self-Ask보다 더 많은 토큰을 사용합니다. 이러한 증가된 토큰 사용량은 GPT-4와 같은 독점 모델로 구현할 경우 더 높은 비용으로 이어질 수 있습니다.

5. 결론

우리는 전문 데이터셋 내에서 지식 추출 및 근거 형성을 개선하기 위해 설계된, 지식 인식 이중 재작성 및 추론 기능으로 강화된 고급 RAG 시스템을 제시한다. 광범위한 실험의 포괄적 결과는 특히 다중 단계 질문이 포함된 벤치마크 시나리오에서 본 접근법의 효용성을 입증한다. 향후 연구에서는 문맥 기반 학습(Wei et al., 2022)을 통합하여 질의 제안자에게 적합한 데모를 적응적으로 선택함으로써 시스템의 숙련도를 개선할 계획입니다. 이는 지식 인식형 질문 재작성 능력을 한층 강화할 것입니다. 또한, 샘플 질문으로부터의 피드백을 통합할 수 있는 지식 인식 원자화기 개발에 관심을 두고 있으며, 이를 통해 가장 유익한 유형의 원자적 지식에 대한 이해도를 향상시킬 수 있을 것입니다.

영향력 진술

본 접근법은 기존 대규모 언어 모델을 활용하여 추가 훈련을 피하고 새로운 편향성 도입을 최소화하며, 사전 처리된 지식 기반에서 응답을 생성해 신뢰성을 보장합니다. 이 과정은 질문 분해의 각 단계를 기록하여 투명하고 해석 가능한 추론 체인을 생성하며, 민감한 환경에서 데이터 보안을 강화하기 위해 비공개로 배포될 수 있습니다. 이 접근법은 법률 연구, 의료 진단, 기술 지원과 같은 분야에서 검색 강화 생성(RAG) 기술의 활용을 발전시켜 의사 결정의 질과 효율성을 향상시킵니다. 정보의 명확성, 정확성 및 논리적 일관성이 강화되면 더 나은 의료 결과, 더 정확한 법률 판단 및 개선된 기술 지원으로 이어져 사회 복지와 발전에 크게 기여할 수 있습니다.

참고문헌

Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

아사이, A., 우, Z., 왕, Y., 실, A., 하지시르지, H. Self-rag: 자기 성찰을 통한 지식 기반 학습의 검색, 생성 및 비판 학습, 2023. URL https://arxiv.org/ abs/2310.11511.

Bender, E. M., Gebru, T., McMillan-Major, A., and Mitchell, M. 확률적 앵무새의 위험성에 관하여: 언어 모델은 너무 커질 수 있는가? *2021 ACM 공정성, 책임성 및 투명성 컨퍼런스( ) 논문집*, pp. 610–623. ACM, 2021.

버틀러, U. 오픈 오스트레일리아 법률 Q&A, 2023. URL https://huggingface.co/datasets/umarbutler/open-australian-legal-qa.

Cao, S., Zhang, J., Shi, J., Lv, X., Yao, Z., Tian, Q., Li, J., and Hou, L. 지식 집약적 복잡한 질문에 답하기 위한 확률적 사고 트리 추론. *arXiv preprint arXiv:2311.13982*, 2023.

Chen, T., Wang, H., Chen, S., Yu, W., Ma, K., Zhao, X., Zhang, H., and Yu, D. Dense x retrieval: What retrieval granularity should we use? *arXiv pr eprint arXiv:2312.06648*, 2023. URL https://arxiv.org/pdf/2312.06648.pdf.

Cohere. 관련 없는 검색 결과에 작별을 고하세요: Co-here 재순위 지정 기능이 출시되었습니다. https://txt.cohere.com/rerank/, 2023. 접속일: 2023-08-28.

Edge, D., Trinh, H., Cheng, N., Bradley, J., Chao, A., Mody, A., Truitt, S., and Larson, J. 지역적에서 전역적 접근: 질의 중심 요약에 대한 그래프 네트워크 접근법, 2024. URL https://arxiv.org/abs/2404.16130.

Fang, Y. 외. 다중 홉 질문 응답을 위한 계층적 그래프 네트워크. *계산언어학회(ACL) 연례 회의 논문집*. 계산언어학회, 2020.

Fei, Z., Shen, X., Zhu, D., Zhou, F., Han, Z., Zhang, S., Chen, K., Shen, Z., and Ge, J. Lawbench: 대규모 언어 모델의 법률 지식 벤치마킹. *arXiv pr eprint arXiv:2309.16289*, 2023.

Feng, Z., Feng, X., Zhao, D., Yang, M., and Qin, B. 검색-생성 시너지 강화 대규모 언어 모델, 2023. URL https://arxiv.org/abs/2310.05149.

Gao, L., Ma, X., Lin, J., and Callan, J. 관련성 라벨 없이 정밀한 제로샷 밀도 검색, 2022. URL https://arxiv.org/abs/2212.10496.

Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., and Wang, H. 대규모 언어 모델을 위한 검색 강화 생성: 개요. *arXiv 사전 인쇄본 arXiv:2312.10997*, 2023.

Hayashi, H., Budania, P., Wang, P., Ackerson, C., Neer-vannan, R., and Neubig, G. WikiAsp: 다중 도메인 측면 기반 요약 데이터셋. *Transactions of the Association for Computational Linguistics*, 9: 211–225, 2021.

Ho, X., Nguyen, A.-K. D., Sugawara, S., and Aizawa, 추론 단계의 포괄적 평가를 위한 다중 홉 QA 데이터셋 구축. *arXiv 사전 인쇄본 arXiv:2011.01060*, 2020.

Jiang, Z., Xu, F. F., Gao, L., Sun, Z., Liu, Q., Dwivedi-Yu, J., Yang, Y., Callan, J., and Neubig, G. 능동적 검색 강화 생성, 2023. URL https://arxiv. org/abs/2305.06983.

Jiapeng, L., Runze, L., Yabo, L., Tong, Z., Mingling, L., and Xiang, C. 리뷰 트리: 다중 홉 질문 답변을 위한 트리 기반 동적 반복 검색 프레임워크, 2024. URL https://arxiv.org/abs/ 2404.14464.

Khattab, O., Santhanam, K., Li, X. L., Hall, D., Liang, P., Potts, C., and Zaharia, M. Demonstrate-search-predict: 지식 집약적 NLP를 위한 검색 및 언어 모델의 조합, 2023. URL https://arxiv.org/abs/2212.14024.

Kojima, T., Gu, S. S., Reid, M., Matsuo, Y., and Iwasawa, Y. 대규모 언어 모델은 제로샷 추론기이다. *신경 정보 처리 시스템의 진전*, 35: 22199–22213, 2022.

루이스, P., 페레즈, E., 픽투스, A., 페트로니, F., 카르푸킨, V., 고얄, N., 쿠틀러, H., 루이스, M., 이, W., 록타셸, T., 외. 지식 집약적 자연어 처리 작업을 위한 검색 강화 생성. *Advances in Neural Information Processing Systems*, 33:9459–9474, 2020.

Li, R. and Du, X. 설명 가능한 다중 홉 질문 응답 및 추론을 위한 구조화된 정보 활용, 2023. URL https://arxiv.org/abs/2311.03734.

Liang, L., Sun, M., Gui, Z., Zhu, Z., Jiang, Z., Zhong, L., Qu, Y., Zhao, P., Bo, Z., Yang, J., Xiong, H., Yuan, L., Xu, J., Wang, Z., Zhang, Z., Zhang, W., Chen, H., Chen, W., and Zhou, J. Kag: 지식 증강 생성을 통한 전문 분야에서의 학습 모델 강화, 2024. URL https://arxiv.org/abs/2409.13731.

Ling, C., Zhao, X., Lu, J., Deng, C., Zheng, C., W ang, J., Chowdhury, T., Li, Y., Cui, H., Zhang, X., Zhao, T., Panalkar, A., Mehta, D., Pasquali, S., Cheng, W., Wang, H., Liu, Y., Chen, Z., Chen, H., White, C., Gu, Q., Pei, J., Yang, C., and Zhao, L. Domain specialization as the key to make large language models disruptive: A comprehensive survey, 2024. URL https://arxiv.org/abs/2305.18703.

마, 셴, 공, 옌, 허, 펑, 자오, 허, 두안, 녜. 검색 강화형 대규모 언어 모델을 위한 쿼리 재작성. *arXiv 사전 인쇄본 arXiv:2305.14283*, 2023.

Panda, P., Agarwal, A., Devaguptapu, C., Kaul, M., and P, P. A. Holmes: 다중 홉 질문 답변을 위한 초관계형 지식 그래프와 언어 모델 활용, 2024. URL https://arxiv.org/abs/2406.06027.

Press, O., Zhang, M., Min, S., Schmidt, L., Smith, N. A., and Lewis, M. 언어 모델에서 구성성 격차 측정 및 축소, 2023. URL https://arxiv.org/abs/2210.03350.

Qiu, M. 외 다수. 다중 홉 질문 답변을 위한 재귀 신경망의 동적 융합. *자연어 처리의 경험적 방법론 컨퍼런스(EMNLP) 논문집*. 계산언어학회, 2019.

Raina, V. and Gales, M. 기업용 RAG를 위한 원자 단위 기반 질문 중심 검색, 2024. URL https://arxiv.org/abs/2405.12363 .

Ram, O., Levine, Y., Dalmedigos, I., Muhlgay, D., Shashua, A., Leyton-Brown, K., and Shoham, Y. In-context retrieval-augmented language models. *Transactions of the Association for Computational Linguistics*, 11:1316–1331, 2023.

Shao, Z., Gong, Y., Shen, Y., Huang, M., Duan, N., and Chen, W. 반복적 검색-생성 시너지를 통한 검색 강화형 대규모 언어 모델 향상, 2023. URL https://arxiv.org/abs/2305.15294.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozie`re, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., and Lam-ple, G. Llama: 개방적이고 효율적인 기초 언어 모델. *arXiv 사전 인쇄본 arXiv:2302.13971*, 2023.

Trivedi, H., Balasubramanian, N., Khot, T., and Sabharw al, A. Musique: 단일 홉 질문 조합을 통한 다중 홉 질문. *계산 언어학 협회 논문집*, 10:539–554, 2022.

Trivedi, H., Balasubramanian, N., Khot, T., and Sabharw al, A. 지식 집약적 다단계 질문을 위한 사고 과정 추론과 검색의 병렬 처리, 2023. URL https://arxiv.org/abs/2212.10509.

왕, C., 류, X., 옌, Y., 탕, X., 장, T., 지아양, C., 야오, Y., 가오, W., 후, X., 치, Z., 왕, Y., 양, L., 왕, J., 시에, X., 장, Z., 장, Y. 대규모 언어 모델의 사실성 조사: 지식, 검색 및 도메인 특이성, 2023a.

Wang, Y., Lipka, N., Rossi, R. A., Siu, A., Zhang, R., and Derr, T. 다중 문서 질문 응답을 위한 지식 그래프 프롬프팅, 2023b. URL https://arxiv. org/abs/2308.11730.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q. V., and Zhou, D. Chain-of-thought prompting elicits reasoning in large language models. Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A. (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 24824–24837. Curran Associates, Inc., 2022.

Xu, S., Pang, L., Shen, H., Cheng, X., and Chua, T.-S. Search-in-the-chain: 지식 집약적 작업을 위한 검색을 통한 대규모 언어 모델의 대화형 향상. *ACM 웹 컨퍼런스 2024 논문집*, pp. 1362–1373, 2024.

Yang, S. Advanced rag 01: Small-to-big re-trieval. https://towardsdatascience.com/advanced-rag-01-small-to-big-retrieval-172181b396d4, 2023. 접속일: 2023-08-28.

Yang, Z., Qi, P., Zhang, S., Bengio, Y., Cohen, W. W., Salakhutdinov, R., and Manning, C. D. Hotpotqa: A dataset for diverse, explainable multi-hop question an-swering. *arXiv preprint arXiv:1809.09600*, 2018.

Zhang, J., Zhang, H., Zhang, D., Liu, Y., and Huang, S. End-to-end beam retrieval for multi-hop question answering, 2024a. URL https://arxiv.org/abs/ 2308.08973.

장, K., 젱, J., 멍, F., 왕, Y., 쑨, S., 바이, L., 쉔, H., 저우, J. 대규모 언어 모델을 활용한 복잡한 질문 응답을 위한 추론 트리 기반 질문 분해. *인공 지능 학회 AAAI 학술대회 논문집*, 38(17):19560–19568, 2024b.

정, H. S., 미슈라, S., 천, X., 청, H.-T., 치, E. H., 레, Q. V., 주, D. 한 걸음 물러서기: 대규모 언어 모델에서 추상화를 통한 추론 유도. *arXiv 사전 인쇄본 arXiv:2310.06117*, 2023.

정, H. S., 미슈라, S., 천, X., 성, H.-T., 치, E. H., 레, Q. V., 주, D. 한 걸음 물러서기: 대규모 언어 모델에서 추상화를 통한 추론 유도, 2024. URL https://arxiv.org/abs/2310.06117.

Zhong, Z., Liu, H., Cui, X., Zhang, X., and Qin, Z. Mix-of-granularity: 검색 강화 생성을 위한 청크 분할 세분성 최적화, 2024. URL https://arxiv.org/abs/2406.00456.

A. 부록

부록 A.1은 세 가지 오픈 도메인 벤치마크에 대한 상세한 소개를 제공하며, 부록 A.3은 실험에 사용된 하이퍼파라미터를 열거하고, 부록 A.4는 포괄적인 실험 결과를 제시합니다.

본 방법론에 대한 보다 심층적인 이해를 위해 부록 A.5에는 비용 분석이 수록되어 있습니다. 또한, 이 하위 섹션에서는 대체 원자 태그 표현에 대한 소개와 논의도 제시합니다.

부록 A.6에서는 세 가지 실제 사례 연구를 탐구합니다. 본 접근법의 네 가지 구성 요소 전반에 걸쳐 사용된 프롬프트는 부록 A.7에 개요가 제시되며, 분해 시연의 영향에 대한 논의가 함께 수록되어 있습니다. 마지막으로, 두 가지 법적 벤치마크에 대한 평가 결과는 부록 A.8에 상세히 기술되어 있습니다.

A.1. 개방형 도메인 벤치마크 소개

본 실험에 사용된 다중 홉 QA 데이터셋에 대한 간략한 개요를 제시하며, 본 방법론은 이러한 분류에 무관하도록 설계되었기에 문제 유형 정보나 홉 수 정보를 해결 과정에서 활용하지 않음을 밝힌다. 표 4는 샘플링된 데이터셋 내 문제 유형 분포를 요약하여 평가에 제시된 다양한 추론 과제에 대한 통찰을 제공하나, 이는 본 방법론에 직접적인 영향을 미치지 않는다.

HotpotQA HotpotQA 데이터셋은 주로 2단계 질문으로 구성된 잘 알려진 다단계 QA 벤치마크로, 각 질문은 10개의 위키백과 문단과 연결됩니다. 이 중 일부 문단은 질문에 답하는 데 필수적인 지원 사실을 포함하는 반면, 나머지는 방해 요소 역할을 합니다. 이 데이터셋에는 *질문* 유형 필드도 포함되어 있어 필요한 논리적 추론 방식을 구분합니다. *비교* 질문은 두 개체를 대조하는 것을, *연결* 질문은 연결 개체를 추론하거나 중간 개체를 통해 개체의 속성을 추론하거나 답변 개체를 찾는 것을 요구합니다(Yang et al., 2018). 본 방법론은 이러한 유형과 독립적으로 작동하지만, 여기서 유형을 설명하는 것은 데이터셋 내 문제의 특성을 예시하고 다양한 벤치마크 간 예상 성능 차이를 맥락화하기 위함이다.

2WikiMultiHopQA HotpotQA에서 영감을 받은 2WikiMultiHopQA는 질문 유형의 다양성을 확장합니다. HotpotQA의 *비교* 유형을 유지하면서, 각각 엔티티 속성 추론과 엔티티 위치 파악에 초점을 맞추어 브릿지 유형에서 진화한 *추론* 및 *구성* 질문을 도입합니다. 또한 *브릿지 비교* 유형은 *브릿지* 추론과 *비교* 추론을 종합적으로 요구하는 새로운 범주입니다. 이 데이터셋은 일반적으로 2-hop에서 4-hop 질문을 제시하며, 각 질문에는 지원 사실과 방해 요소를 포함한 10개의 위키백과 문단이 동반됩니다. 이러한 유형들은 데이터셋 구조를 알려주지만, 본 방법론에서는 활용되지 않습니다. 본 방법론은 분류와 무관하게 모든 질문을 균일하게 처리합니다. 간결성을 위해 본 논문에서는 2WikiMultiHopQA를 2Wiki로 약칭합니다.

MuSiQue는 다중 단계 질문 중 상당수가 단축 경로로 해결될 수 있다는 문제점—적절한 추론 없이 정답에 도달하는 현상—을 해결하기 위해, Trivedi 등(Trivedi et al., 2022)이 보고한 바와 같이 연결된 추론을 장려하도록 특별히 설계된 엄격한 필터와 추가 메커니즘을 구현합니다. 다른 데이터셋과 달리 MuSiQue는 질문 유형별로 분류하지 않지만, 각 질문에 필요한 홉 수(2~4홉)에 대한 명시적 정보를 제공합니다. 각 질문에는 20개의 문맥 단락이 연결되어 있으며, 여기에는 관련 정보와 무관한 정보가 혼합되어 있어 올바른 추론 경로를 식별하는 작업을 더욱 복잡하게 만듭니다. 이러한 명시적인 홉 정보는 본 방법론에서 사용되지는 않지만, 데이터 세트의 복잡성과 이러한 과제를 효과적으로 처리하기 위해 모델에 요구되는 견고성을 강조합니다.

Table 4. 세 가지 다중 홉 QA 데이터셋에 걸친 질문 유형 분포.

유형	카운트	비율
비교	107	21.4%
교량	393	78.6%

(a) HotPotQA

유형	개수	비율
비교	132	26.4%
추론	64	12.8%
구성적	196	39.2%
교량 비교	108	21.6%

(b) 2WikiMultiHopQA

홉 수	개수	비율
2	263	52.6%
3	169	33.8%
4	68	13.6%

(c) MuSiQue

A.2. 평가 방법 소개

표 1에 제시된 방법 외에도, 반복적 탐색 및 분해 방법인 SearChain(Xu et al., 2024)과 지식 그래프 기반 방법인 GraphRAG(Edge et al., 2024)를 활용한 실험을 수행하였다. GraphRAG는 로컬 모드와 글로벌 모드 모두에서 추론되었다. 본 연구에서 평가된 방법은 다음과 같다:

- **Zero-Shot CoT**: 질문은 오직 Chain-Of-Thought(CoT) 기법만을 사용하여 처리됩니다. 이 기법은 예시 시연이나 보충적 맥락의 도움 없이 LLM이 단계별로 추론 과정을 설명하도록 유도합니다. 이 방법은 제로샷 환경에서 LLM의 내재적 지식과 추론 능력을 평가합니다.
- **순진한 RAG(Naive RAG)**: 평면 지식베이스에서 밀도 검색(dense retrieval)을 수행하여 각 질문에 대한 관련 정보를 획득하는 접근법입니다. 지식베이스는 사전 임베딩된 정보 조각들로 구성되며, 의미적 유사성을 기반으로 원본 질문과 매칭됩니다. 검색 과정은 중간 작업 분해 없이 직접적으로 이루어집니다.
- **자기 질문 및 검색** 기반: 이 방법은 작업 분해 전략을 활용하여 LLM이 반복적으로 후속 질문을 생성하고 답변하도록 유도함으로써 복잡한 문제를 관리 가능한 하위 작업으로 분해합니다. 모든 벤치마크에 대해 작업 분해의 논리와 방법론을 설명하는 일반적인 시연을 제공하여 LLM의 추론 과정을 안내합니다. 기존 설정(Press et al., 2023)과 달리, 본 설정에서는 후속 질문에 대한 답변을 LLM 자체 지식에만 의존하지 않고 추가적인 검색 구성 요소를 도입합니다. 참조 컨텍스트를 제공하기 위해 평면 지식베이스에서 후속 질문을 쿼리로 사용하여 관련 정보 블록을 검색합니다. 또한 다른 방법론과 일관성을 유지하기 위해 분해 과정을 최대 *N개의* 후속 질문으로 제한합니다.
- **IRCoT**: 이 접근법은 LLM에게 검색된 문단과 함께 한 문장 더의 근거를 생성하도록 반복적으로 프롬프트하고, 새로 생성된 근거로 새로운 문단을 검색합니다. 원본 설정은 최대 토큰 수로 프로세스를 제한합니다(Trivedi et al., 2023). 본 실험에서는 총 반복 횟수를 본 방법에 사용한 *상수 N으로* 제한합니다.
- **Iter-RetGen**: 이 방법은 검색된 문단으로 질문에 반복적으로 답변하며, 새로 생성된 근거와 답변을 다음 라운드 검색에 활용합니다. 이 설정에서도 총 반복 횟수를 *동일한 N으로* 제한합니다.
- **SearChain**: 이 접근법은 LLM과 정보 검색(IR) 간의 상호작용에 중점을 둡니다. SearChain은 각 노드가 IR 지향적 질의-답변 쌍으로 구성된 LLM 생성 추론 체인인 Chain-of-Query(CoQ)에서 시작합니다. 이후 CoQ의 각 노드 답변을 IR을 통해 반복적으로 검증하고, 검색된 정보와 일치하지 않는 노드에 대해 CoQ를 재생성합니다. 이러한 재생성 메커니즘을 통해 SearChain은 트리 기반의 새로운 추론 경로를 형성하여 LLM이 추론 방향을 동적으로 수정할 수 있게 합니다. 공식 코드는 사전 훈련된 모델을 온라인에 업로드하지 않고 로컬에서 불러오므로, 우리는 HuggingFace에서 가장 유사한 이름의 모델을 찾아 이를 적용합니다. 또한 환경적 문제로 인해 표 6에 제시된 실험 결과는 ColBERT 리트리버 대신 *BAAI/bge-m3로* 수행되었습니다.
- **ProbTree**: 이 접근법은 명시적 트리 탐색 방법입니다. ProbTree는 주어진 질문에 대해 LLM이 번역한 쿼리 트리에서 시작하며, 여기서 각 비루트 노드는 상위 노드의 하위 질문을 나타냅니다. 그런 다음 질문 분해와 답변 모두의 신뢰도를 고려하여 잎에서 루트로 질문을 해결함으로써 트리 위에서 확률적 추론을 수행합니다.
- **GraphRAG Local**: 공개 지침에 따라 지식 그래프를 구축하기 위해 지식베이스를 전처리합니다. 평가는 로컬 모드에서 추론됩니다.
- **GraphRAG Global**: 공개 지침에 따라 지식 그래프를 구축하기 위해 지식베이스를 전처리합니다. 평가는 글로벌 모드에서 추론됩니다.
- **KAR^3 (저희)**: 제안된 지식 인식 분해 방법은 복잡한 질문을 하위 질문으로 반복적으로 분해하고 최대 N회 반복까지 관련 지식을 검색합니다. 이 과정은 최종 답변의 맥락을 가장 유용한 다섯 개의 지식 단위로 제한합니다.

논의된 평가 방법 간의 차이를 보다 명확히 설명하기 위해, 각 방법의 특징을 체계적으로 정리한 표 5를 제시합니다. 이 표는 질문 분해, 지식 조각 검색, 그리고

표 5. 방법 비교.

방법	분해			검색	생성 컨텍스트	
	증명	경로	컨텍스트		하위 답변	최종 답변
제로샷 CoT 순진한		해당		해당 없음	해당 없음	해당 없음
RAG 자체 질문 w/		없음		질문 → 청크	해당 없음	청크 질문-
R. IRCoT	소량 데이터	해당	qa 쌍 근거, 청크 청크	하위 질문 → 청크 근거 문장 → 청크 전체 근거 → 청크 하위 질문 → 청크	청크 해당	답변 쌍
Iter-RetGen	학습 소량	없음	∅; qa 쌍, 청크		없음 해당	근거, 청크 청크
SearChain	데이터 학습		∅	하위 질문 → 청크	없음	질의-응답
ProbTree	제로샷 소량	체인 (암시	선택된 청크	하위 질문 → 원자 질문 → 청크	∅; 청크	쌍 질의-응
KAR^3 **(우리의)**	데이터 학습	적) (암시적)			청크	답 쌍
	소량 데이터	체인 트리			해당 없음	선택된 청크
	학습 제로샷	동적				

답변 생성에 사용되는 컨텍스트를 설명합니다. 구체적으로 각 방법이 제로샷 또는 소량 데이터 학습 조건에서 작동하는지, 분해 과정의 특성(예: 명시적 또는 암시적 분해; 체인형, 트리형 또는 동적으로 생성된 경로), 그리고 분해 과정에서 활용되는 컨텍스트를 구분합니다. 검색 열은 각 방법이 정보를 수집하기 위해 사용하는 메커니즘을 명확히 설명하며, 하위 답변 및 최종 답변 생성을 위한 생성 컨텍스트 전용 열은 각 방법이 답변을 생성할 때 활용하는 특정 컨텍스트를 강조합니다.

표 5에서 보여지듯, KAR^{3}의 분해 모듈은 제로샷 지식 인식 접근법을 채택하여 반복적 분해를 위해 축적된 선택된 청크를 문맥 내에서 유지합니다. 또한 부록 A.7에서는 데모스트레이션 통합의 잠재적 이점을 논의하며, 이 기능이 성능을 더욱 향상시킬 수 있음을 시사합니다. 이 가능성은 향후 탐구를 위해 예정되어 있습니다. 특히, 본 접근법은 반복 과정에서 분해 경로를 동적으로 구성하여 문맥적으로 제공된 지식으로부터 얻은 새로운 통찰에 기반한 조정이 가능하도록 합니다. 검색 단계에서는 원자 태그를 활용하여 쿼리와 청크 내 정보 간의 의미적 간극을 해소합니다. 중요한 점은 생성 단계에서 본 방법이 선택된 청크를 유지함으로써, 생성 과정이 지식 인식 상태를 유지하고 오직 후속 질문과 답변에 의존하는 방법에서 흔히 발생하는 오류 누적 위험을 완화한다는 것입니다.

A.3. 하이퍼 파라미터

지식 추출 단계에서는 지식 분해 과정에 특화된 0.7의 *온도* 설정을 활용하여 생성된 원자 지식의 다양성과 결정론적 특성 간 균형을 촉진합니다. 반면 각 방법의 모든 질문-답변(QA) 단계에서는 *온도를* 0으로 설정하여 모델의 일관된 응답을 보장합니다.

검색 구성 요소와 관련하여, 일반 지식베이스와 원자 지식베이스 모두에 대해 텍스트 임베딩 모델로 *text-embedding-ada-002*(버전 2)를 사용합니다. Naive RAG 및 Iter-RetGen에 사용되는 일반 지식베이스의 경우, 검색기는 검색 점수 임계값 0.2를 적용하여 최대 16개의 지식 청크를 가져오도록 구성됩니다. Self-Ask w/ Retrieval 및 IRCoT에서 사용되는 일반 지식베이스의 경우, 검색된 청크가 단일 후속 질문 답변이나 단일 연속적 근거 문장 생성에 사용되므로 전체 근거 또는 최종 질문 답변을 위한 참조 청크가 누적됩니다. 시스템은 요청당 4개의 관련 청크를 검색하며, 동일한 점수 임계값 0.2를 유지합니다. 원자 지식베이스의 경우, 검색기는 각 원자 쿼리에 대해 4개의 관련 원자 태그를 검색하도록 설정되어 있지만, 콘텐츠 길이가 짧기 때문에 더 높은 임계값 0.5를 적용합니다.

A.4. 상세한 실험 결과

평가 지표 평가 지표와 관련하여 부록에서 세 가지 추가 지표를 활용합니다. 응답이 사전 정의된 정답과 동일한지 평가하는 **정확 일치도(EM)**는 커뮤니티에서 일반적으로 사용해 온 방식대로 적용됩니다. 또한, 특정 방법이 높은 정확도(Acc) 점수를 달성하면서도 낮은 F1 점수를 기록하는 상황을 종종 접하게 됩니다. 이러한 불일치의 근본 원인을 규명하기 위해 생성된 응답의 **재현율(Recall)**과 **정밀도(Precision)**도 함께 보고합니다. 리콜은 응답에 포함된 정답 라벨의 관련 토큰 비율을 측정하는 반면, 정밀도는 생성된 응답 내 토큰이 정답 라벨과 얼마나 관련성이 높은지를 평가합니다.

표 6. 다중 단계 질의응답 데이터셋에 대한 상세 성능 비교. 최상위 성능은 굵게, 차상위 성능은 밑줄 처리.

표 6. (a) HotPotQA

방법	EM	F1	정확도	정밀도	재현율
제로샷 CoT	32.60	43.94	53.60	46.56	43.97
순진한 RAG	56.80	72.67	82.60	74.52	74.86
자기질문 및 회상	57.00	71.40	80.00	73.25	73.95
IRCoT	51.40	67.30	81.00	69.32	72.15
Iter-RetGen	59.60	75.27	86.60	77.18	77.62
SearChain	28.60	40.48	74.40	40.77	66.63
ProbTree	47.00	62.41	73.40	64.83	64.95
GraphRAG 로컬	0.00	10.66	**89.00**	5.90	**83.07**
GraphRAG 글로벌	0.00	7.42	64.80	4.08	63.16
KAR^3 (우리)	**61.40**	**76.48**	88.00	**78.53**	78.96

표 6. (b) 2WikiMultiHopQA

방법	EM	F1	정확도	정밀도	재현율
제로샷 CoT	35.67	41.40	43.87	41.43	43.11
순진한 RAG	51.20	59.74	62.80	59.06	62.30
자기 질문과 회상	60.60	69.06	75.00	67.88	73.15
IRCoT	55.00	63.83	70.40	62.47	68.86
Iter-RetGen	57.80	67.21	73.60	66.10	71.09
SearChain	7.00	15.67	68.40	11.91	66.74
ProbTree	57.00	69.42	80.00	67.61	76.89
GraphRAG 로컬	0.00	11.83	71.20	6.74	75.17
GraphRAG 글로벌	0.00	7.35	45.00	4.09	55.43
KAR^3 (우리)	**65.80**	**75.00**	**82.20**	**73.63**	**79.08**

표 6. (c) MuSiQue

방법	EM	F1	정확도	정밀도	재현율
제로샷 CoT	12.93	22.90	23.47	24.40	24.10
순진한 RAG	32.00	43.31	44.40	44.42	47.29
자기 질문 및 회상	38.20	46.76	51.40	46.75	51.00
IRCoT	36.00	47.57	49.20	48.70	50.30
Iter-RetGen	40.20	52.48	55.60	53.51	56.45
SearChain	24.40	33.26	45.80	33.00	46.37
ProbTree	28.57	43.26	52.86	42.27	54.70
GraphRAG 로컬	0.60	9.62	49.80	5.73	55.82
GraphRAG 글로벌	0.00	5.16	44.60	2.82	52.19
KAR^3 (우리)	**47.40**	**57.86**	**62.60**	**58.52**	**61.37**

상세 주요 결과 다중 홉 데이터셋 HotpotQA, 2Wiki 및 MuSiQue에 대한 상세 실험 결과는 표 6에 제시되어 있습니다. 표 1에 표시된 지표 외에도 **EM**, **정밀도** 및 **재현율이** 여기에 제공됩니다.

그래프 기반 방법에 대한 논의 특히 지식 그래프 기반 방법인 GraphRAG Local은 2-hop 질문이 주를 이루는 HotpotQA 데이터셋에서 탁월한 성능을 보입니다. 그러나 더 많은 홉을 포함하는 질문이 포함된 다른 두 데이터셋에서는 GraphRAG Local이 IRCoT와 동등한 수준에 그칩니다. 이는 지식 그래프 기반 방법이 복잡한 다중 홉 질문을 해결하는 데 직면한 어려움을 강조합니다. GraphRAG의 경우, (Edge et al., 2024)에서 제시된 쿼리 중심 요약(QFS) 작업을 위해 설계되었으나, 본 방법에 비해 로컬 및 글로벌 모드 모두에서 성능이 미흡함을 관찰했다. GraphRAG는 흥미로운 경향을 보인다: 정확도와 리콜 점수는 높게 달성하는 반면, EM, F1, 정밀도 지표에서는 낮은 성능을 보인다. GraphRAG 출력을 자세히 분석해 보면, 쿼리를 반복하고 답변에 대한 메타정보를 그래프 구조 내에 포함시키는 경향이 드러납니다. QA 프롬프트를 개선하려는 시도에도 불구하고 이 행동은 지속됩니다. 표 7은

HotpotQA의 질문에 대한 GraphRAG Local의 응답을 보여주는 대표적인 예시입니다.

표 7. HotpotQA 질문에 대한 GraphRAG 로컬 출력 예시. 이 표는 질문을 반복하고 응답에 메타 정보를 포함하는 경향을 보여줍니다.

질문	알사 몰과 스펜서 플라자가 위치한 국가는 어디인가요?
답변 레이블	인도
GraphRAG의 답변	알사 몰과 스펜서 플라자는 모두 인도 첸나이에 위치해 있습니다. [데이터: 인도 및 첸나이 커뮤니티 (2391); 엔티티 (4901, 4904); 관계 (9479, 1687, 5215, 5217)].

N 선택에 대한 상세 평가 결과 표 8은 반복 *상한 N에* 대한 제거 연구를 위해 그림 5에 제시된 항목에 따른 세부 성능 지표를 나열합니다. 답변 레이블의 리콜 토큰을 나타내는 표 6의 리콜과 달리, 여기서 **리콜***은 해당 데이터셋이 제공하는 지원 사실의 리콜을 나타냅니다.

표 8. 하이퍼파라미터 *N*에 대한 제거 연구. Recall*은 지원 사실의 리콜을 나타냅니다.

N	HotpotQA			2Wiki			MuSiQue		
	리콜*	F1	정확도	리콜*	F1	정확도	리콜*	F1	Acc
1	42.96	59.46	70.20	40.41	41.08	43.00	31.20	32.55	32.80
2	82.04	74.27	84.80	78.83	70.22	77.20	56.43	48.46	50.00
3	90.16	76.90	87.20	87.71	72.84	79.40	64.82	53.50	57.20
4	92.46	76.49	87.80	92.86	74.68	81.80	69.87	55.73	59.40
5	92.83	76.48	88.00	94.06	75.00	82.20	73.08	57.86	62.60
6	93.35	77.67	89.00	94.76	75.12	81.80	74.88	57.03	61.20
7	93.68	77.32	88.80	94.91	75.44	82.40	76.07	56.66	61.40
8	93.78	76.88	88.40	95.06	75.16	82.00	76.72	57.65	62.40
9	93.78	76.99	88.60	95.11	74.89	81.80	76.90	57.17	61.40
10	93.78	77.52	89.00	95.16	75.09	82.00	77.20	57.69	62.40

덜 발전된 LLM을 사용한 평가 결과 한계 논의 섹션에서 소개한 바와 같이, 우리는 GPT-3.5를 활용하여 일련의 실험을 수행했습니다. 이 실험 결과는 표 9에 정리되어 있습니다. 이 특정 실험에서는 언어 모델로 GPT-4(1106-Preview) 대신 GPT-3.5(1106-Preview)를 사용했으며, 표 1에 요약된 실험과 동일한 모든 실험 설정을 유지했습니다.

표 9. GPT-3.5를 활용한 구현체 성능 비교. 최상위 결과는 굵게, 차상위 결과는 밑줄 표시.

방법	HotpotQA		2Wiki		MuSiQue	
	F1	정확도	F1	Acc	F1	Acc
회상을 통한 자기 질문	49.52	61.40	53.83	**60.00**	31.05	35.20
IRCoT	56.39	<u>68.40</u>	40.31	46.00	33.93	34.40
Iter-RetGen	48.63	66.80	44.32	55.20	25.77	<u>37.80</u>
KAR^3 (우리)	46.37	**68.80**	41.95	<u>58.20</u>	26.80	**39.60**

A.5. 비용 분석 및 논의

본 절에서는 모델의 API 사용량을 평가하기 위한 포괄적인 비용 분석을 수행합니다. 먼저 추론 비용을 다른 기준 방법과 비교 평가하고, 이후 비용을 구성 요소로 세분화하며, 마지막으로 일회성 데이터 전처리 단계의 비용 요약 정보를 제공합니다.

추론 비용 비교 표 10에서 보여준 바와 같이, QA당 토큰 소비 측면에서 본 방법은 ProbTree 및 IRCoT보다 적은 토큰을 사용하며 Iter-RetGen과 유사한 수준입니다. 그러나 본 접근법은 F1 점수와 정확도 모두에서 이러한 기준 모델들을 상당한 차이로 크게 능가합니다. 이는 비용과 성능 균형 측면에서 본 방법의 효율성을 입증합니다.

비용과 성능의 균형을 효과적으로 조율함을 입증합니다. 특히 본 방법은 잠재적 추론 체인 탐색에 중점을 두어, 각 반복 단계에서 문맥을 고려한 신중한 질문 분해 분석이 필요하다는 점을 강조할 필요가 있습니다. 그 결과, 완성 토큰 사용량이 전체 소비량의 약 1/4을 차지하며, 이는 본 접근법을 다른 기준선과 차별화하는 특징입니다.

표 10. MuSiQue에서의 토큰 소비량(평균/QA) 및 성능 비교.

방법	토큰 소비량 (⇓)			성과 (⇑)	
	프롬프트	완료율	총합	F1	정확도
제로샷 CoT	85	105	191	22.90	23.47
순진한 RAG	1765	103	1869	43.31	44.40
자기질문 및 회상	5894	619	6514	46.76	51.40
IRCoT	9703	86	9789	47.57	49.20
Iter-RetGen	8140	473	8614	52.48	55.60
ProbTree	25225	650	**25875**	43.26	52.86
KAR^3 (우리)	6525	2295	8820	**57.86**	**62.60**

다양한 구성 요소의 토큰 소비량 지금까지 제시된 실험 결과에서는 디컴포저, 셀렉터, 제너레이터 구성 요소에 동일한 대규모 언어 모델(LLM)이 사용되었습니다. 해당 구성 요소를 위해 설계된 프롬프트는 부록 A.7에 상세히 기술되어 있습니다. 이러한 구성 요소가 서로 다른 언어 모델을 사용하도록 구성될 수 있다는 점은 주목할 만하며, 이는 향후 연구 과제로 남겨둡니다. MuSiQue에서 각 구성 요소의 상세한 토큰 소비량은 표 11에 설명되어 있습니다. 분해-선택 루프는 최대 5회 반복되며, 이는 각 QA에 대해 분해기와 선택기에 대한 다중 호출로 이어집니다. 결과적으로 분해기와 선택기가 전체 소비량의 대부분을 차지합니다.

표 11. MuSiQue에서의 토큰 소비량 (평균/QA).

구성 요소	프롬프트	완성	총
쿼리 제안자	2691	768	3459
원자 선택자	3278	1429	4707
답변 생성기	556	98	654
KAR^3 (우리)	6525	2295	8820

토큰 소비량 챕크 원자화 챕크 원자화는 일회성 전처리 단계로, LLM API 소비량은 데이터 챕크 수에 선형적으로 비례하며 벤치마크별로 약간씩 차이가 나는 오버헤드를 구성합니다. 4.1절에서 설명한 바와 같이, 모든 청크는 컨텍스트 단락에서 파생되며, 청크 수에 상응하는 LLM 호출 횟수는 토큰 소비량과 함께 표 12의 마지막 열에 참고용으로 기재되어 있습니다. 입력 토큰 크기(표에서 프롬프트)는 주로 청크 크기에 의해 결정되는 반면, 출력 토큰 크기(표에서 완성)는 생성된 원자 태그의 크기에 따라 달라집니다.

표 12. 토큰 소비량(평균/청크) 및 청크 수 통계.

데이터셋	프롬프트	완료	총	호출
HotpotQA	209	129	338	4950
2Wiki	199	122	321	3410
MuSiQue	197	123	320	7120

원자 태그의 대체 표현 방식 대규모 데이터셋에 본 방법을 적용할 때 확장성의 중요성을 인지합니다. 확장성을 유지하면서 비용 효율성을 높이기 위해 Llama 3와 같은 오픈소스 언어 모델을 통합하여 전처리 비용을 크게 절감합니다. 또한 자원 사용을 추가로 최적화하기 위해 대체 원자 태그 표현 방식을 탐구합니다. 유망한 접근법 중 하나는 데이터를 일반 텍스트 문장으로 분해하여 각 문장을 원자 태그로 취급하는 것입니다. 이 방법은 *spacy* 라이브러리를 활용해 원본 데이터 청크를 문장으로 분할함으로써 사전처리 단계를 단순화하고, 언어 모델 호출의 필요성을 제거합니다. 표 13에 상세히 기술된 평가 결과에 따르면, 이 접근법은 MuSiQue 데이터셋에서 성능이 55.2%로 감소했음에도 불구하고 여전히 대부분의

기준 방법들. 이는 저비용 전처리가 우선순위인 시나리오에서 잠재적 효과성을 입증하며, 비용과 성능을 효율적으로 균형 잡는 실행 가능한 대안을 제시한다.

표 13. MuSiQue에서 대체 원자 태그의 성능.

LLM	원자 태그	F1	정확도
Llama 3	일반 텍스트 문장	45.88	54.20
	원자적 질문 (우리의)	**50.68**	**59.70**
GPT-4	일반 텍스트	50.72	55.20
	원자적 질문 (우리의)	**57.86**	**62.60**

A.6. 실제 사례 연구

본 섹션에서는 제안된 분해 파이프라인의 기본 원리를 설명하기 위해 평가 벤치마크에서 추출한 세 가지 실제 사례 연구를 제시합니다. 이는 알고리즘 1에 상세히 기술된 바와 같습니다. 이러한 실제 사례를 통해 체계적인 접근 방식의 장점을 부각하고자 합니다. 각 사례는 파이프라인의 각 단계가 성능 향상에 어떻게 기여하는지, 그리고 구현 과정에서 얻은 통찰력을 조명할 것입니다.

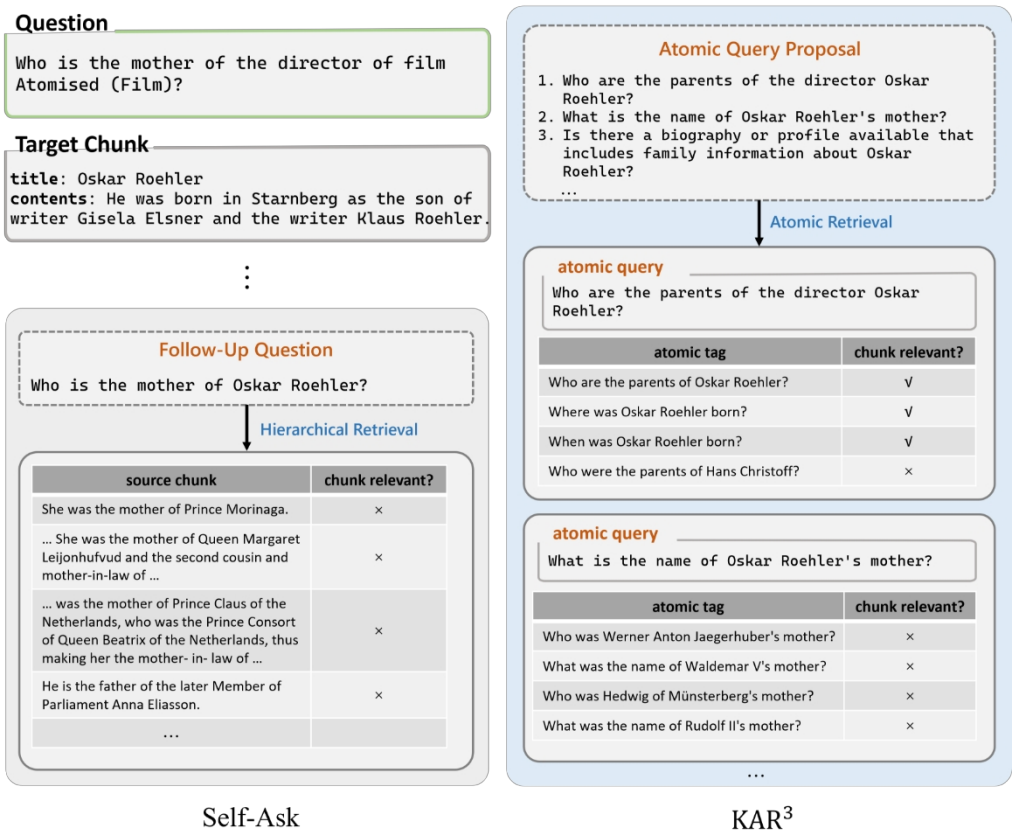


그림 6. 사례 (a): 대중적인 영화 "What Women Want"에 비해 덜 알려진 영화 "What Women Love"를 예로 들면, 왼쪽의 Self-Ask와 같은 단일 경로 방식은 후자에 대한 후속 질문을 생성하는 경향이 있어 최종 답변이 잘못될 수 있다. 반면 KAR³은 여러 원자적 질의를 가정하여 원본 질문의 의도된 의미를 효과적으로 식별하고, 관련 원자 태그를 제공하여 작업 이해를 원자 선택 단계로 연기한 후 정확한 결론에 도달할 수 있다.

KAR³은 원본 질문의 의도된 의미를 효과적으로 식별함으로써 단일 경로 방식보다 우수한 성능을 발휘합니다. 우리의 작업 분해 전략은 Self-Ask 접근법에서 보여준 것처럼 단일 결정론적 후속 질문을 생성하기보다 다중 원자적 질의를 생성하는 것을 포함합니다. 현대적 분해 방법들은 일반적으로 생성 모델을 활용하여 단일 후속 질문을 구성합니다. 그러나 이 접근법은 오류가 있는 질문을 생성할 수 있는 내재적 위험을 안고 있어, 잘못된 분해 경로로 이어지고 궁극적으로 오류 답변을 초래할 수 있습니다. 그림 6에 묘사된 사례 (a)를 고려해 보십시오. 여기서 원본 질문은 "What Women Love"라는 영화에 관한 것입니다. 더 유명한 영화 "What Women Want"가 존재하기 때문에, 사용된 언어 모델은 원본 질문을 '수정'하려는 경향이 있습니다. 결과적으로 Self-Ask(그림 6 좌측 예시)와 같은 방법은 이 잘못된 대상에 대한 단 하나의 후속 질문만 생성합니다. 그림에서처럼 임베딩 유사성으로 인해 대상 챕터는 검색되었으나, '잘못된' 후속 질문에 대한 '잘못된' 중간 답변이 생성되어 최종적으로 잘못된 응답이 도출됩니다. 반면, 우리의 방법론은 "What Women Love"와 "What Women Want" 모두에 관한 원자적 질의를 제기함으로써 초기 질문의 진정한 의도를 명확히 하려 합니다. 두 영화가 존재하고 관련 원자적 태그가 검색되면, 우리의 접근법은 이후 원자적 선택 단계에서 질문 의도를 검증하고 정확하고 가장 적절한 청크를 선택할 수 있는 이점을 얻습니다.

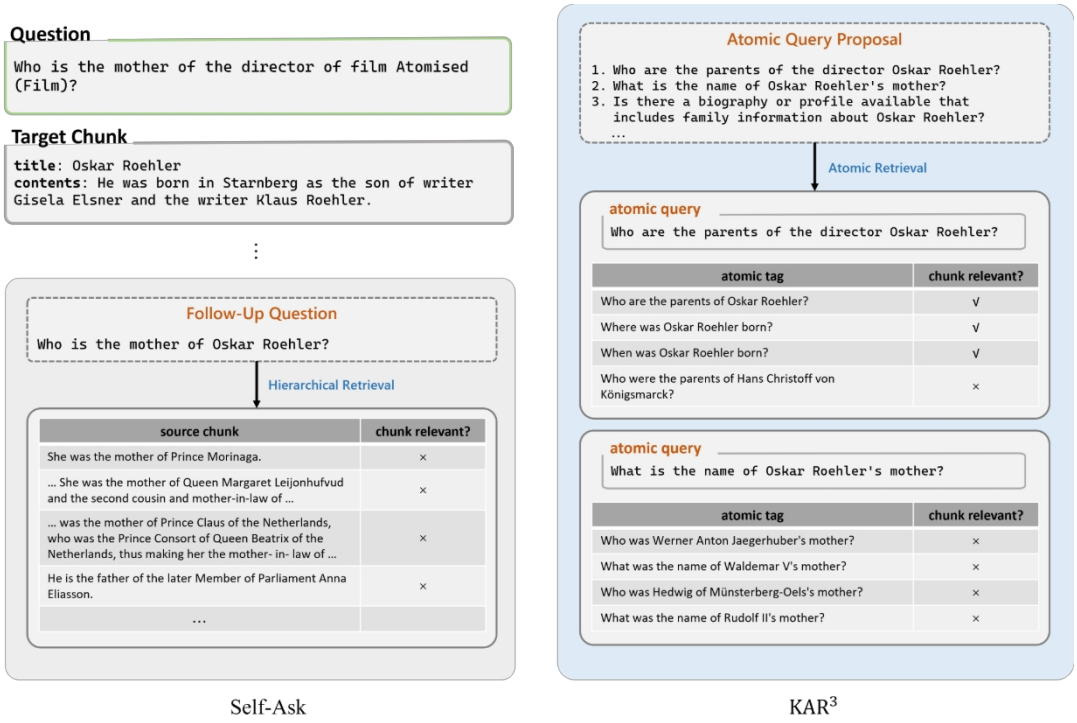


그림 7. 사례 (b): KAR³은 다중 원자 쿼리를 제안함으로써 관련 지식 청크를 효과적으로 검색하는 반면, Self-Ask가 채택한 단일 결정론적 후속 질문 접근법은 지식 기반의 스키마와 정렬되지 않아 검색 실패를 초래한다.

KAR³은 원자 태그를 매개체로 삼아 지식 스키마 정렬을 개선함으로써 기존 기법을 능가합니다. 코퍼스와 질의의 표현 방식 간 불일치는 단일 결정론적 접근보다 다중 질의 방식을 지지하는 또 다른 핵심 요소입니다. 표현 격차는 생성된 후속 질의가 의미론적으로 정확하더라도 검색 과정을 방해할 수 있습니다. 예를 들어, 그림 7의 사례 (b)에서 보듯이, 왼쪽의 Self-Ask와 같은 단일 경로 방식은 '오스카 뢰러의 어머니는 누구인가?'라고 직접 질문할 수 있습니다. 그러나 지식베이스는 가족 관계를 다른 스키마('A는 B와 C의 아들이다')로 표현하므로, 질문이 정확함에도 불구하고 검색 과정이 실패합니다. 계층적 검색을 Self-Ask에 적용했을 때조차도, Self-Ask with Hierarchical Retrieval은 이 간극을 메우지 못했습니다. 반면, 다중 원자 쿼리를 생성하는 우리의 접근법은 지식베이스의 다양한 표현에 대응하는 더 넓은 범위의 표현을 포괄합니다. 제시된 사례에서 오스카 로엘러의 어머니를 직접 묻는 원자 쿼리는 동일한 검색 문제를 겪지만, 그의 부모에 대한 정보를 찾는 대체 쿼리는 목표 정보 블록을 성공적으로 검색합니다. 이는 쿼리 생성의 유연성이 지식베이스 구조와의 정렬 가능성을 높이는 방식을 보여줍니다.

정확한 정보를 얻는 것.

우리의 방법론은 덩어리를 직접 검색하기보다 원자 태그를 검색하는 것을 강조합니다. 이러한 설계 선택은 그림 7에 묘사된 사례 (b)에서 잘 드러납니다. 코퍼스의 지식 덩어리는 'A는 B와 C의 아들이다'라는 패턴으로 구조화되어 있어 '...의 어머니는 누구인가'와 같은 쿼리로 직접 검색하기 어렵습니다. 우리의 전문 지식베이스에서는 이러한 직접적인 질의가 'A는 B의 어머니이다' 또는 'A는 B의 아버지이다' 패턴에 부합하는 덩어리를 검색하는 경향이 있습니다. 검색을 위한 중간 매개체로 원자 태그를 활용함으로써, 우리의 접근법은 단일 질의와 지식베이스 내 다중 문장 구조 사이의 간극을 효과적으로 좁힙니다. 이는 본 사례에서 '~의 어머니' 대 '~의 아들'로 대표되는 표현 패턴 차이를 연결하는 데 기여합니다.

KAR³은 간결하고 관련성이 높은 문맥을 유지함으로써 중간 답변에 의존하는 방법들보다 우수한 성능을 보입니다. 후속 처리를 위해 중간 답변만 보존하는 Self-Ask와 같은 방법과 달리, 본 방법은 전체 청크를 문맥 정보로 보존합니다. 원자 선택 단계에서, 우리는 원본 청크의 관련 내용에 대한 후보 요약으로 원자 태그 목록을 제시합니다. 이 전략은 토큰 사용량을 크게 줄이고 관련 정보 선택 과정을 단순화합니다. 그림 8의 사례 (c)는 본 접근법의 이중적 이점을 보여줍니다: 첫째, 선별된 원자 태그 목록에서 선택함으로써 관련 정보 식별을 효율화합니다; 둘째, 중간 답변만 유지하는 대신 선택된 전체 청크를 보존함으로써 정확하고 포괄적인 후속 처리를 위한 풍부한 맥락을 보장합니다. 왼쪽의 Self-Ask 방식은 대상 청크를 검색하지만 과도한 문맥 정보로 인해 관련 'Ernie Watts'를 정확히 식별하지 못합니다. Self-Ask에서 검색된 청크는 중간 답변 생성 후 폐기되므로, 이 방법은 잠재적으로 잘못된 경로를 따라 부정확한 결론에 도달할 수 있습니다. 반면, 우리의 접근법은 간결한 목록에서 적절한 원자 태그를 효율적으로 필터링하고 선택할 수 있습니다. 이번 라운드의 원자 태그는 어니 와츠의 역할과 관련되지만, 그의 출생지에 대해 추가로 질의할 필요가 없습니다. 이 정보는 선택된 청크 내에 포함되어 있으며, 후속 라운드에서 맥락으로 활용될 수 있도록 유지되기 때문입니다.

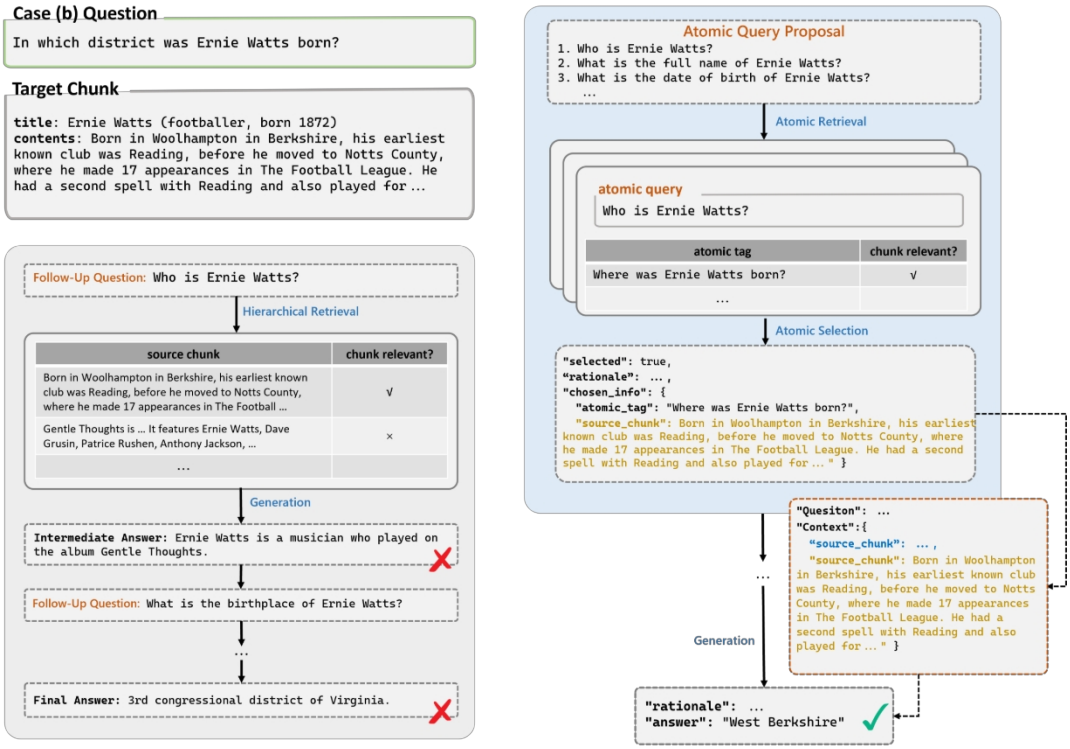


그림 8. 사례 (c): KAR³은 대상 선택을 위한 간결한 원자 태그 목록을 활용하는 동시에 풍부한 문맥 지원을 위해 전체 청크를 유지한다는 장점이 있다. 반면 Self-Ask의 접근법은 관련 청크 검색에는 성공하지만, 문맥을 위해 중간 답변에 의존하는 한계로 인해 결국 잘못된 최종 답변을 생성하게 된다.

A.7. 프롬프트 설계

본 접근법은 네 가지 별개의 프롬프트를 사용한다: (1) 원자적 질문 태깅 프롬프트: 소스 단락을 전처리할 때 사용되며, 각 단락을 여러 원자적 질문과 원자적 태그로 연결한다; (2) 원자적 질의 제안 프롬프트: 다중 원자적 질의 제안을 생성할 때 사용되며, 알고리즘 1의 1행을 참조한다; (3) 원자 태그 선택 프롬프트: 주어진 질문 목록에서 가장 유용한 원자 태그를 선택할 때 사용되며, 알고리즘 1의 1번 라인을 참조함; (4) 질문 답변 프롬프트: 분해 루프 종료 시 주어진 질문에 대한 최종 답변을 생성하기 위해 적용되며, 알고리즘 1의 1번 라인에 설명됨.

원자적 질문 태깅 프롬프트

```
# 과제
주어진 콘텐츠로 답변 가능한 관련 질문을 최대한 많이 추출하는 것이 당신의 과제입니다. 다양성을 유지하고 중복되거나 유사한 질문 추출
을 피하십시오.
중복되거나 유사한 질문은 피하십시오. 질문에는 필요한 명사형 표현을 포함하고, '그것', '그', '그녀', '그들', '회사', '사람' 등과 같은 대
명사 사용을 피하십시오.

# 출력 형식
각 질문을 새 줄에 하나씩, 항목 구분 기호나 번호 없이 출력하십시오.

# 콘텐츠
{content}

# 출력
```

원자적 쿼리 제안자 프롬프트

```
# 작업
제공된 맥락을 분석한 후, 질문에 더 잘 답할 수 있도록 도움을 줄 수 있는 지식에 대해 원자적인 하위 질문들을 제기하십시오. 다양한 방식으로
사고하고 가능한 한 많은 다양한 질문들을 제기하십시오.

# 출력 형식
다음 JSON 형식으로 출력하십시오:
{{
     "thinking": <이 작업에 대한 당신의 사고 과정, 질문과 주어진 상황에 대한 분석 포함>,
     "sub questions": <필요한 내용을 나타내는 하위 질문 목록>
}}

# 컨텍스트
이미 확보된 컨텍스트:
{선택된 내용}

# 질문
{content}

# 출력 결과
```

원자 태그 선택 프롬프트

```
# 작업
당신의 임무는 제공된 맥락을 분석한 후, 주어진 질문에 답하기 전에 먼저 답변하는 것이 유용할 수 있는 하위 질문들을 결정하는 것입니다. 주어진 질문 목록에서 가장 관련성이 높은 하위 질문을 선택하십시오. 주어진 맥락이나 본인의 지식으로 이미 답변 가능한 하위 질문은 선택하지 마십시오.
관련성 높은 하위 질문을 선택하십시오. 주어진 맥락이나 본인의 지식으로 이미 답할 수 있는 하위 질문은 선택하지 마십시오.

# 출력 형식
다음 JSON 형식으로 출력하십시오:
{{
    "thinking": <이 선택 작업에 대한 당신의 생각>,
    "question idx": <하위 질문 인덱스, 1부터 {num atom questions} 사이의 정수>
}}

# 컨텍스트
이미 가지고 있는 문맥:
{선택한 내용}

# 선택 가능한 하위 질문들
{atom 질문 목록 문자열}

# 질문
{내용}

# 출력 결과
```

질문 답변 프롬프트

```
# 작업
주어진 맥락(있는 경우)을 참조하여 질문에 답하십시오. 마지막 질문에 답하기 위해서는 먼저 제공된 기사, 보고서 또는 맥락을 읽은 후 최종 답변을 제시해야 합니다.

# 출력 형식
출력은 반드시 아래 형식을 준수해야 합니다. Python에서 json으로 파싱 가능하도록 출력하십시오.
{{
    "answer": <답변, 문자열 형식으로 작성>, "rationale": <선택 이유 설명>
}}

# 컨텍스트 (해당 시)
{컨텍스트 (있는 경우)}

# 질문
{질문 내용}{제한 없음}

단계별로 생각해보자.
```

시연 기반 **논의** 현재 실험에서는 모든 프롬프트가 제로샷 방식으로, 즉 기대되는 추론 논리를 설명하는 시연이 제공되지 않습니다. 시연이 성능 향상에 기여할 수 있는지 탐구하기 위해 제거 연구를 설계했습니다. 기존에 사용된 Self-Ask w/ Retrieval 및 IRCoT 방법론을 적용하여 프롬프트와 작업 설명을 수정함으로써, 이러한 방법들의 제로샷 방식이며 시연이 없는 변형 버전을 생성했습니다. 이를 **'검색 기반 제로샷** 셀프-어스크**(Zero-Shot Self-Ask w/ Retrieval)'**와 **'제로샷 IRCoT(Zero-Shot IRCoT)'**로 명명했습니다. 실험 결과는 표 14에 제시되어 있습니다. 실험 결과에 따르면, '검색 기반 제로샷 셀프-어스크' 방법은 2Wiki 및 MuSiQue 데이터셋에서 정확도가 소폭 하락했는데, 이는 생성 과정의 본질적인 무작위성 때문일 수 있습니다. 그러나 데모를 포함하면 모든 F1 점수가 크게 향상되고 IRCoT 방법의 전반적인 성능이 개선됩니다. 이는 단계별 분해 접근법에 의존하는 방법에 데모가 특히 유용할 수 있음을 시사합니다. 따라서 데모 통합은 KAR³프레임워크 내에서 향후 연구를 위한 유망한 방향으로 확인되었습니다.

표 14. 성능 비교: 제로샷 대 퓨샷.

방법	HotpotQA		2Wiki		MuSiQue	
	F1	정확도	F1	Acc	F1	Acc
검색을 통한 제로샷 자기질문	55.76	76.20	54.98	76.20	40.97	50.40
검색 기능이 포함된 자체 질문	71.40	80.00	69.06	75.00	46.76	51.40
제로샷 IRCoT	58.22	75.80	49.69	60.20	37.17	43.00
IRCoT	67.30	81.00	63.83	70.40	47.57	49.20

A.8. 법률 벤치마크 평가

이 하위 섹션에서는 LawBench(Fei et al., 2023)와 Open Australian Legal QA(Butler, 2023)라는 두 가지 법률 벤치마크에서 본 접근법의 성능을 제시합니다. 이를 진행하기 전에 각 벤치마크에 대한 간략한 설명을 제공합니다.

LawBench LawBench는 중국 법률을 위한 포괄적인 법률 벤치마크입니다. 이는 대규모 언어 모델(LLM)의 법률 역량을 정확히 평가하기 위해 세심하게 설계된 20개의 과제로 구성됩니다. 일부 기존 벤치마크가 객관식 문제에만 의존하는 것과 달리, LawBench는 실제 적용과 밀접하게 연관된 다양한 유형의 과제를 포함합니다. 이러한 과제에는 법인체 인식, 독해력 평가, 범죄 금액 계산, 법률 상담 등이 포함됩니다. 모든 과제가 RAG(질문-응답-추론) 중심은 아니므로(예: 독해), 6가지 특정 과제를 선정하였으며, 이는 표 15에 상세히 기술되어 있습니다. 각 과제의 문제 수는 500개입니다.

표 15. LawBench 작업 개요

과제 번호	과제	유형	지표
1-1	법령 낭독	생성	F1
1-2	법률 지식 Q&A	단일 선택	EM
3-1	법률 예측 (사실 기반)	복수 선택	EM
3-2	법령 예측 (시나리오 기반)	생성	F1
3-6	사례 분석	단일 선택	EM
3-8	상담	생성	F1

또한 독자들의 참고를 위해 이러한 작업의 예시 질문을 제공합니다(GPT-4를 사용하여 번역됨).

1-1 : 다음 질문에 기사 내용을 직접 제시하여 답변하십시오:
→ 증권법 제76조의 내용은 무엇입니까?
1-2 : '증권법'에 따르면, 주식
→ 거래소에 관한 다음 진술 중 틀린 것은 무엇입니까? A: 증권거래소의 허가 없이는 어떤 기관도
→ 개인 또는 기관은 실시간 증권 거래 정보를 공개할 수 있습니다; B: 증권 거래소는
→ 거래소는 필요 시 주요 비정상적 거래 조건을 보이는 증권 계좌에 대한 거래를 제한할 수 있으며, 증권 규제 당국에 보고할 수 있다.
거래 조건을 보인 증권 계좌에 대해 필요에 따라 거래를 제한하고, 증권 규제 당국에 보고할 수 있다
C: 회원제 증권거래소의 축적된 재산은 회원에게 속하며, 그 권리는 회원들이 공동으로 향유한다.
→ 증권거래소의 축적된 재산은 회원에게 귀속되며, 회원들은 공동으로 권리를 향유한다.
→ 회원; 존속 기간 동안 축적된 재산은 회원에게 분배될 수 없다.
D: 증권거래소는 상장규정, 거래규정, 회원관리규정 및 기타 증권법에 따른 관련 규정을 제정한다.
→ 관리 규정 및 기타 관련 규정을 증권법 및
→ 행정 규정에 따라 증권 거래소는 상장 규정, 거래 규정, 회원 관리 규정 및 기타 관련 규정을 제정하고
국무원( )에 기록을 위해 보고한다.
3-1 : 다음 사실과 혐의에 근거하여 관련 조항을 제시하십시오.
형법. 사실관계: 길림성 유수시 피고인은 2015년 11월 15일
→ 2015년 11월 15일, 피고인 허씨는 국씨(국씨)가 소유한
→ 차량(번호판 xxx)에 대한 월세 3,900위안(월납)을 정했다. 2016년 1월 19일, 구오 씨의 모르게
→ 3,900.00 위안으로 매월 지급하기로 했다. 2016년 1월 19일, 구오 씨의 알지 못하는 사이에 피고인 허 씨는
→ 피고인은 사실을 숨기고 자신이 택시의 소유주라고 거짓으로 주장했다.
→ 그는 피해자 마 씨와 월 임대료 3,800위안, 임대 기간 1년의 차량 임대 계약을 체결하고 총 50,600위안을 징수했다.
→ 3,800.00위안, 임대 기간 1년으로 계약하고, 마 씨로부터 총 50,600.00위안의
2016년 2월 26일, 해당 택시는
→ 피해자 마로부터 소유주 구오가 회수하였다. 피해자 마는 피고 허에게 임대료와 보증금을 반환할 것을 반복하여 요구했으나, 피고 허는 반환을 거부하였다.
→ 피고인 허에게 임대료와 보증금을 반환해 달라고 요구했으나, 피고인 허는 이를 거부했다.

→ 피해자 진술서, 증인 진술서, 서면 증거 등을 제시하였으며, 피고인 허 씨가

표 16. 법적 벤치마크에 대한 평가 결과 (표 15에 명시된 대로 지표는 **F1 / EM**)

과제		제로샷 CoT	GraphRAG 로컬	우리 모델 (N=5)
로벤치	1-1	21.31	23.27	**78.58**
	1-2	54.24	62.60	**70.60**
	3-1	53.32	74.60	**83.16**
	3-2	27.51	25.98	**46.05**
	3-6	51.16	47.64	**61.91**
	3-8	17.44	18.43	**23.58**
오픈 오스트레일리아 법률 QA		25.10	34.35	**63.34**

표 17. 법률 벤치마크에 대한 평가 결과 (지표는 **정확도**)

작업		제로샷 CoT	GraphRAG 로컬	우리 방법 (N=5)
로벤치	1-1	1.23	16.60	**90.12**
	1-2	54.00	63.40	**70.60**
	3-1	49.90	75.40	**88.82**
	3-2	15.83	27.60	**67.54**
	3-6	51.12	57.00	**62.73**
	3-8	49.70	58.80	**61.72**
오픈 오스트레일리아 법률 QA		16.48	88.27	**98.59**

피고인 허씨는 불법 점유를 목적으로, 서명 과정에서 사실을 조작하고 진실을 은폐함으로써 타인의 재산을 사기적으로 취득하였다.
→ 계약 체결 및 이행 과정에서 사실을 조작하고 진실을 은폐하여 타인의 재산을 사취하였음.
→ 계약 체결 및 이행 과정에서 사실을 조작하고 진실을 은폐함으로써 타인의 재산을 사취하였다. 해당 금액은 상대적으로 거액이었으며, 그의 행위는
→ 중화인민공화국 형법 제xx조 규정을 위반하였으며
‹ → 계약 사기죄로 형사 책임을 물어야 한다.
사기.

3-2 : 구체적인 시나리오와 질문에 따른 법적 근거를 제시해 주십시오.
→ 특정 법조문의 내용만 필요하며, 각 시나리오에는

→ 하나의 법률 조항만 포함됩니다. 시나리오: 화물선이 하역 항구에 도착했으나
수하인이 제때 도착하여 화물을 인수하지 못한 경우. 어떤 법적
선장은 다른 적절한 장소에서 화물을 하역할 수 있는가?
3-6: 바가 개업한 지 1년 후, 사업 환경이 급변했고 모든
→ 파트너들은 대책을 논의하기 위해 회의를 열었다. '합자기업법'에 따르면
→ 기업법에 따르면, 다음의 투표 사항은 유효한 투표로 간주됩니다: A: 장
‹ → '통청'이라는 이름이 매력적이지 않다고 생각하여 '
→ 통성 바.' 왕과 조는 동의하지만 리는 반대한다; B: 부진한 사업 상황을 고려하여
→ 사업 부진을 고려하여 왕은 한 달간 영업을 중단하고 리모델링 및
장 씨와 조 씨는 동의하지만 리 씨는 반대함; C: 바의 긴급한 필요로 인해
→ 바의 긴급한 필요로 인해, 조는 바에 커피 머신 일괄 판매를 제안한다. 장과
왕은 동의하지만 리는 반대한다; D: 네 명의 파트너가 법률 사무소 운영 경험이 부족하다는 점을 고려하여
→ 관리 경험이 부족하다는 점을 고려하여, 리는 자신의 친구 왕을 관리 파트너로 임명할 것을 제안한다. 장
장 씨와 왕 씨는 동의하지만, 조 씨는 반대한다.

3-8: 거주자 A가 B에게 집을 임대했다. A의 동의하에 B는 임대 주택을 리모델링한 후
C가 단독으로 주택의 하중 지지 구조를 변경하였다.

→ 왜 A는 B에게 계약 위반에 대한 책임을 지도록 요구할 수 있나요?

오픈 오스트레일리아 법률 QA 벤치마크는 오스트레일리아 법률 코퍼스에서 GPT-4가 합성해낸 2,124개의 질문과 답변으로 구성됩니다. 모든 질문은 생성형 유형입니다. 예시: "뉴사우스웨일스 주 앤더슨 대 아미티지 사건 [2014] NSWCATCD 157에서 법 제63조에 따른 임대인의 일반적 의무는 무엇인가?"

평가 결과는 표 16에 정리되어 있으며, 여기서 우리는 "GraphRAG Global"보다 일반적으로 이러한 작업에서 더 우수한 성능을 보이는 "GraphRAG Local"과만 비교합니다.

상기 이유로, 우리는 모든 실험 결과를 평가하기 위해 GPT-4를 사용하며 정확도(**Acc**)를 표 17에 보고합니다. 표 16과 표 17의 결과를 비교해 보면, 일부

지표가 크게 변했음에도 결과의 순서는 유지되는 것을 관찰할 수 있습니다. 다음 섹션에서는 이러한 변화의 원인을 규명하고자 하며,

이는 향후 RAG 프레임워크 평가를 위한 더 나은 지표 설계에 유용한 통찰력을 제공할 수 있을 것입니다.

1. 생성 작업(1-1, 3-2, Open Australian Legal QA)에서 본 접근법의 정확도는 크게 향상됩니다. 이러한 작업에서 우리의 답변은 종종 의미론적으로는 동등하지만 구문적으로는 골든 답변과 다릅니다. 이는 GPT-4가 답변의 의미적 내용을 비교할 수 있기 때문에 지표 성능이 개선된 이유를 설명합니다. 이는 "Open Australian Legal QA" 작업에 대한 "GraphRAG Local" 결과에도 적용됩니다.

2. 생성 작업 1-1과 3-2에서는 "GraphRAG Local"의 정확도가 하락합니다. 해당 작업은 법률 조항 인용 및 예측을 포함하며 특정 조항 검색이 필요합니다. 상세 분석 결과, "GraphRAG Local"은 종종 올바른 조항을 검색하지 못하거나 잘못된 조항을 참조하지만, 법률 정보를 반복하는 경향이 있음을 확인했습니다. 따라서 법률 명칭과 "XX 법률에 따르면, XX 조항..."과 같은 일반적인 접두사를 재구성하는 것만으로도 토큰 수준 리콜을 개선할 수 있습니다.

3. 우리의 접근법과 "GraphRAG Local" 모두 작업 3-8에서 상당한 정확도 향상을 보입니다. 첫 번째 요점에서 언급된 이유 외에도, 골든 답변의 품질도 이러한 차이에 기여할 수 있습니다. 과제 3-8의 질문과 골든 답변은 컨설팅 웹사이트에서 수집되었기 때문에 품질이 제각각입니다. 예를 들어, 한 질문은 "원혼인에서 태어난 자녀들이 아버지를 부양할 의무가 있는가?"라고 묻습니다. 그러나 제공된 골든 답변에는 미성년 자녀에 대한 부모의 부양 의무를 다루는 관련 없는 조항인 "제1067조"가 포함되어 있습니다.

질문: 양부모가 모두 이혼하여 각자 새로운 가정을 꾸린 경우
↛새로 아이를 둔 가정이며, 법원 판결에 따르면 아버지는
↛ 자녀가 만 18세가 될 때까지 매월 양육비를 어머니에게 지급해야 합니다.
원래 결혼에서 태어난 자녀들은 아버지를 부양할 의무가 있나요?
아버지를 부양할 의무가 있습니까?

참조 답변: 우리나라에서는 친자녀가 이혼한 부모를 부양할 의무가 있습니다.
↔자녀와 부모의 관계는
↔자녀나 부모의 이혼으로 인해 해산되지 않는다.
자녀의 부모 부양은 법적 의무이다. 자녀가 노부모를 부양하지 않을 경우
↛노인을 부양하지 않을 경우, 부모는 인민법원에 직접 소송을 제기할 수 있으며
법적 근거: 제
↔중화인민공화국 민법 제1067조는 부모가
↛부양의무를 이행하지 않을 경우, 미성년 자녀 또는 독립생활이 불가능한 성인 자녀는
↛ 독립적으로 생활할 수 없는 경우 부모에게 부양을 청구할 권리가 있다.
성년 자녀가 부양의무를 이행하지 않을 경우,
↛무능력자이거나 생계에 곤란을 겪는 부모는 부양을 청구할 권리가 있다.
성인 자녀로부터. 제1084조는 부모와 자녀 간의 관계가
↛부모의 이혼으로 인해 부모와 자녀의 관계가 소멸하지 않는다.
↛이혼 후 자녀가 아버지 또는 어머니 중 한쪽에 의해 직접 양육되더라도
↔여전히 양친의 자녀이다. 변호사 설명:
↛부모가 이혼한 후에도 성인 자녀는 여전히 양쪽 부모를 부양할 의무가 있습니다.
성인 자녀가 부모를 부양할 의무는
↛부모의 관계 변화로 인해 달라지지 않습니다. 성인 자녀가
↔ 부양 의무를 이행하지 않을 경우, 무능력하거나 생계에 어려움을 겪는 부모는
↔생계에 어려움을 겪는 부모는 성인 자녀에게 부양을 요청할 권리가 있습니다. 답변: 예, 부양 의무가 있습니다. 법적 근거:
↛중화인민공화국 민법 제1069조에는 다음과 같이 규정되어 있습니다:
↔ '자녀의 부모 부양 의무는
↛ 부모의 혼인 관계 변화로 인해 종료되지 않는다.' 따라서, 설령
↛부모가 이혼하고 새 배우자와 재혼하여 새 자녀를 두더라도, 원래 자녀들은
↛여전히 부모를 부양할 의무가 있다.

4. 선택 과제 1-2, 3-1, 3-6에 대한 모든 방법의 정확도는 예상대로 F1 점수와 거의 일치한다. 예외는 과제 3-1로, 차이는 주로 GPT-4의 중국어 이해 능력, 특히 아라비아 숫자와 한자 숫자 구별 능력에서 비롯된다. 중국 법에서는 모든 숫자를 한자로 표기하지만, 정답 예시에서는 모든 숫자를 아라비아 숫자로 표기했다.