JKSCI

한국컴퓨터정보학회논문지
Journal of The Korea Society of Computer and Information
Vol. 30 No. 11, pp. 19-27, November 2025
https://doi.org/10.9708/jksci.2025.30.11.019

Lightweight DS-RAG: A Training-Free RAG Framework for Multi-Document QA on Edge Devices

Ji-Min Bang*, Woo-Sin Lee**

*Master's Student, Dept. of Computer Information Engineering, Kwangwoon University, Seoul, Korea
**Professor, Dept. of Computer Information Engineering, Kwangwoon University, Seoul, Korea

[Abstract]

Multi-document Question Answering (Multi-document QA) is an essential component of question answering systems in real-world environments. It requires extracting and integrating the necessary information from multiple documents to generate accurate responses. The DS-RAG framework is optimized for real-world multi-document QA tasks. It improves the system's overall accuracy and efficiency by using a document selection module that selects only the core documents retrieved after question decomposition, which are necessary for response generation. However, this process requires additional computational resources and additional training, making it difficult to apply in resource-limited environments. To address this limitation, this study proposes a Lightweight DS-RAG framework that achieves selection effects through importance-weighted query generation, thereby eliminating the need for a separate selection module. The proposed approach identifies key information within a question and generates importance-weighted queries. This enables retrieval focused on the most relevant information while maintaining both efficiency and accuracy. This approach maintains high retrieval accuracy while significantly reducing computational overhead, and can be applied in resource-constrained environments without domain-specific training. Experimental results demonstrate that the Lightweight DS-RAG framework achieves 95% of the retrieval performance of the original DS-RAG, as measured by the F1-score. Additionally, it exhibited an average performance improvement of 67.5% over RAG without a selection module. These results demonstrate that a high level of accuracy can be maintained without a separate selection module or additional training, indicating that the Lightweight DS-RAG framework can serve as a practical alternative in resource-constrained environments.

▸ **Key words**: Retrieval-Augmented Generation(RAG), Multi-document Question Answering, Information Retrieval, Docuent Selection, Generation Optimization, Lightweight Model

• First Author: Ji-Min Bang, Corresponding Author: Woo-Sin Lee
*Ji-Min Bang (bjimin7@kw.ac.kr), Dept. of Computer Information Engineering, Kwangwoon University
**Woo-Sin Lee (woosin.lee@kw.ac.kr), Dept. of Computer Information Engineering, Kwangwoon University
• Received: 2025. 07. 30, Revised: 2025. 09. 03, Accepted: 2025. 10. 03.

http://www.ksci.re.kr pISSN:1598-849X | eISSN:2383-9945

[요 약]

정확한 응답 생성을 위해 다양한 문서로부터 필요한 정보를 추출하고 통합해야 하는 다중 문서 기반 질문 응답(Multi-document QA)은 실제 환경에서의 질문 응답 시스템에 필수적인 요소이다. 이러한 실제 다중 문서 기반 질문 응답에 최적화된 DS-RAG (Dynamic-Selection-based Retrieval-Augmented Generation) 프레임워크는 질문 분해와 검색 과정을 거친 후, 문서 선택(selection) 모듈을 통해 응답 생성에 필요한 핵심 문서만을 선별함으로써 전체 시스템의 정확도와 효율성을 향상시킨다. 그러나 선택 모듈은 별도의 학습 과정과 연산 자원을 요구하기 때문에, 자원이 제한된 환경에는 적용이 어렵다는 한계가 존재한다. 따라서 본 연구는 별도의 선택 모듈 없이도 질문 내 중요도를 반영한 질의 생성을 통해 검색 단계에서 선택 효과를 유도하는 Lightweight DS-RAG 프레임워크를 제안한다. 제안된 방식은 질문 내 핵심 정보를 식별하고, 중요도 기반 질의를 생성하여 주요 정보에 집중된 검색이 가능하도록 설계되었다. 이를 통해 높은 검색 정확도를 유지하면서도 연산 부담을 크게 절감할 수 있으며, 별도의 도메인 특화 학습 없이도 다양한 환경에서 실용적으로 활용될 수 있다. 실험 결과, Lightweight DS-RAG 프레임워크는 F1-score 기준으로 기존 DS-RAG 대비 95% 수준의 검색 성능을 달성하였으며, 선택 모듈이 없는 RAG 대비 평균 67.5% 향상된 성능을 보였다. 이는 별도의 선택 모듈이나 추가 학습 없이도 높은 정답 포함률을 유지할 수 있음을 실험적으로 입증하며, 자원 제약 환경에서 실용적인 대안으로서의 가능성을 보여준다.

▸ **주제어**: 검색 증강 생성(RAG), 다중 문서 질문 응답, 정보 검색, 문서 선택, 생성 최적화, 경량화 모델

I. Introduction

Retrieval-Augmented Generation (RAG) 시스템은 사전 학습된 지식만을 기반으로 응답을 생성하는 언어 모델의 한계를 극복하기 위해 제안된 방식으로, 외부 문서를 검색하고 이를 생성 과정에 통합함으로써 최신 정보나 도메인 특화 지식을 유연하게 반영할 수 있다[1].

이러한 RAG 시스템의 성능은 (1) 검색 단계에서 필요한 정보를 얼마나 정확하게 수집하는지, 그리고 (2) 수집된 정보를 생성 모델에 얼마나 효과적으로 통합하는지에 따라 결정된다[2-3]. 특히 다중 문서 기반의 질문 응답 환경에서는 응답에 필요한 정보가 여러 문서에 분산되어 있기 때문에 필요한 정보가 누락되거나 검색된 정보를 효과적으로 통합하지 못할 경우 응답 정확도가 크게 저하될 수 있다[2-3][6].

초기의 RAG 시스템은 입력 질문 전체를 하나의 단일 질의(query)로 처리하여 문서를 검색한다. 이 경우, 질문 내 다양한 조건이나 복수의 검색 대상이 분리되지 못해 일부 조건에 편향된 검색 결과가 도출되거나, 핵심 문서가 누락되어 불완전한 문서 집합이 생성되는 문제가 발생할 수 있다[2][4].

이러한 한계를 보완하기 위해 질문 분해(Question Decomposition, QD) 기법이 도입되었다[6-7]. 질문 분해는 복합적인 질문을 여러 하위 질문으로 나누어 각 조건과 대상을 개별적으로 처리함으로써 균형 잡힌 정보 검색을 가능하게 한다. 그러나 하위 질문 수가 증가할수록 불필요하거나 상충되는 정보가 포함될 가능성이 높아지며, 이는 생성 품질의 저하 및 입력 길이 증가에 따른 처리 비용 상승으로 이어질 수 있다[2-3][6][13].

이를 완화하기 위해 재정렬(re-rank) 기반의 문서 선택 기법[15-20]이 제안되었으나, 대부분의 재정렬 기법은 연산 비용이 높고 대규모 모델을 요구하기 때문에 실시간성 및 자원 효율성이 중요한 상용 환경에는 적용이 어렵다.

이러한 문제를 해결하기 위해 DS-RAG 프레임워크가 제안되었다[14]. DS-RAG는 비교적 경량화된 선택 모듈을 통해 검색 문서 중 핵심 문서만을 선택함으로써 효율적인 입력 구성을 수행한다. 그러나 이 선택 모듈은 학습 기반 구조로 되어 있어, 도메인별 학습 데이터 구축 및 최적화 과정이 필요하며, 이로 인한 시간·비용·자원 측면의 제약이 존재한다. 이러한 제약은 특히 고성능 모델 활용이 어려운 에지(edge) 디바이스와 같은 자원 제약 환경에서 더

욱 두드러진다.

따라서 본 연구에서는 별도의 학습 없이도 적용 가능한 경량형 RAG 시스템, 즉 Lightweight DS-RAG 프레임워크를 제안한다. 제안된 방식은 (1) 도메인 특화 학습 없이도 높은 검색 정확도를 유지하고, (2) 응답 생성에 필요한 문서만을 선별하여 입력을 최소화함으로써, 구조 기반 질의 생성만으로 기존 DS-RAG와 유사한 성능을 달성함을 실험적으로 검증하였다.

II. Related Works

1. Retrieval-Augmented Generation

RAG는 외부 데이터베이스로부터 문서를 검색하여 이를 생성 모델에 통합함으로써, 사전 학습된 모델이 학습하지 못한 최신 정보나 도메인 지식을 반영할 수 있도록 한 프레임워크이다[1-5]. 이후 다양한 하위 모듈(질문 분해, 문서 재정렬, 문서 선택 등)이 도입되며 구조적 발전이 이루어졌다.

2. Question Decomposition

질문 분해(Question Decomposition, QD) 기법은 복잡하거나 다중 조건을 포함하는 질문을 여러 개의 하위 질문으로 나누어 각 조건이나 대상을 독립적으로 검색할 수 있도록 하는 방법이다[7]. 이를 통해 단일 질의로는 수행하기 어려운 복합적 질문을 보다 정밀하게 분리·처리할 수 있다.

기존 QD 방식은 접근 방식에 따라 학습 기반[7][9][11], 그래프 기반[8][10][12], 대규모 언어 모델(LLM) 기반[6], 규칙 기반 및 구문 분석 방식[11-12][14] 등 다양한 형태로 제안되어 왔다.

그러나 이러한 방식들은 다중 문서 기반 질문 응답 환경에 직접 적용되기에는 최적화되어 있지 않아 다음과 같은 한계가 발생한다. 첫째, 부정확한 분해는 하위 질문 간 의미 충돌이나 정보 누락을 유발하며, 특히 비교형(comparative) 질문에서는 핵심 정보가 손실될 위험이 크다. 둘째, 분해된 질문들이 서로 다른 검색 결과를 유도하여 과도한 수의 문서가 생성될 수 있고, 이는 입력 길이 초과 및 생성 품질 저하로 이어진다. 셋째, 다중 분해로 인해 검색 결과가 중복되거나 상충될 경우, 정답 판단 과정에서 노이즈가 누적되어 최종 응답의 정확성이 저하된다.

따라서 QD는 복잡한 질문을 구조적으로 분리한다는 점에서 유용하지만, 다중 문서 기반 질문 응답 환경에 적용할 경우 검색 효율성과 입력 제약 측면에서 새로운 문제가 발생할 수 있다. 이러한 이유로, 이후 연구에서는 선택(selection) 또는 재정렬(reranking)과 결합하여 QD의 한계를 보완하는 방향으로 발전하고 있다.

3. Reranking

검색 단계에서 얻은 초기 후보 문서 중 실제 응답에 필요한 문서를 선별하기 위해 재정렬(Reranking) 기법이 활용된다[15].

재정렬은 일반적으로 두 단계로 구성된다. 첫 번째 단계는 벡터 검색 기반의 후보 필터링, 두 번째 단계는 정밀 재정렬로, 문맥 이해를 반영하여 핵심 문서를 선별한다.

초기 연구에서는 기계학습 기반 Learning-to-Rank(LTR) 방식을 적용하였다[16]. 이후 트랜스포머(Transformer) 인코더를 활용한 Cross-Encoder 기반 재정렬 모델이 등장하며 정확도가 크게 향상되었다[17]. Cross-Encoder는 질의와 문서를 하나의 입력으로 결합하여 문맥 내 상호작용을 학습함으로써, 문서 간 미세한 의미 차이를 효과적으로 구분한다.

최근에는 대규모 언어 모델(LLM)을 활용한 생성형 재정렬(Generative Reranking) 기법이 주목받고 있다[18-19]. LLM은 추가 학습 없이도 프롬프트를 통해 질의-문서 관계를 추론하거나, “이 문서가 답변에 도움이 되는가?”와 같은 질의응답 형태로 직접 판단을 수행한다. 또한, 일부 연구에서는 기존 방식들을 결합한 하이브리드방식[20]이 제안되어, 정확도와 유연성을 동시에 확보하고자 하였다.

그러나 이러한 재정렬 방식은 공통적으로 높은 연산 비용과 지연(latency)을 수반하며, 도메인 특화 학습 또는 긴 입력 처리 시 비용 증가 문제가 존재한다. 특히 Cross-Encoder는 모든 문서 쌍을 독립적으로 인코딩해야 하므로 효율성이 떨어지고, LLM 기반 재정렬은 추론 품질의 변동성과 속도 저하 문제가 있다. 따라서, 높은 정확도를 유지하면서도 자원 효율성과 실시간성이 요구되는 상용 환경 또는 에지 디바이스에서는 적용이 어렵다.

4. Selection

질문 응답 시스템에서 선택(Selection)은 검색 단계에서 얻어진 문서 집합 중 실제 응답에 필요한 핵심 문서만을 선별하는 과정으로, 다중 문서 기반 환경에서 불필요한 정보 유입을 억제하고 효율적인 생성 입력을 구성하는 핵심 단계이다.

재정렬의 경우 검색된 전체 문서를 대상으로 관련도 순위를 재배열(reorder) 하는 과정으로, 문서의 상대적 중요도를 계산하여 상위 k 개 문서를 결정하는 데 초점을 둔다. 반면, 선택은 응답 생성에 실질적으로 기여하는 문서만을

선택적(subset)으로 추출하는 과정으로, 정확한 정답 포함률을 유지하고 입력 효율을 극대화하는 것이 목표이다.

이러한 선택 과정을 구조적으로 구현한 대표 사례가 DS-RAG 프레임워크[14]이다. DS-RAG은 상업적 다중 문서 기반 질문 응답 환경에 최적화된 구조로, 질문 분해와 문서 선택을 결합하여 필수 정보 중심의 입력 문맥을 구성한다. 질문 분해는 질문의 의미 단위를 명시적으로 구분해 검색 효율을 높이는 보조 역할을 수행하며, 핵심적인 선택 과정은 DICS(Dynamic Input Context Selector) 모듈에서 이루어진다.

DICS는 질문의 그래프 구조를 기반으로 선택 기준(selection criteria)을 정의하고, 검색된 문서 청크 중 해당 기준과 의미적으로 관련된 청크만을 선택한다. 선택 기준은 질문 내 개체와 관계를 임베딩하여 구성된 그래프 표현으로 정의되며, 이 벡터와 문서 청크의 임베딩 간 유사도를 계산해 입력 포함 여부를 결정한다. 이를 통해 DICS는 전체 문서를 재정렬하거나 통합하지 않고도, 질문의 기준점(anchor point)과 직접적으로 연관된 핵심 정보만을 효율적으로 선별한다. 또한, 질문의 유형이나 기준점 수에 따라 입력 크기를 동적으로 조정하여 불필요한 정보를 최소화하고 정답 청크 포함 가능성을 극대화한다.

그러나 이러한 DICS는 선택 기준을 학습하기 위해 도메인 특화 데이터와 추가 학습 과정이 필요하다는 한계를 가진다. 이는 연산 비용이 제한된 환경이나 별도의 학습이 어려운 상황에서는 적용이 제한될 수 있다.

III. The Proposed Scheme

본 연구는 DS-RAG[14]의 선택 모듈(DICS)을 제거하고, 질문 그래프 기반 선택 질의 생성 모듈(GS-QG, Graph-based Selective Query Generator)을 도입하였다. 이를 통해 별도의 학습 없이도 구조적으로 선택 효과를 유도하여 효율적인 문서 검색을 수행한다.

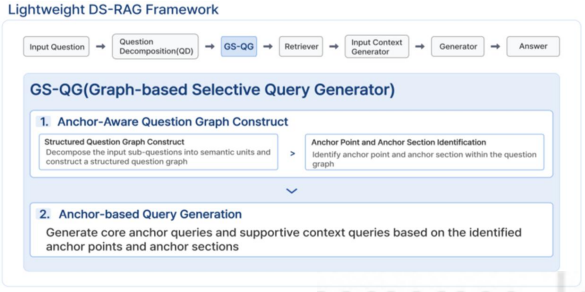

Fig. 1. Lightweight DS-RAG Framework

Fig. 1은 제안한 Lightweight DS-RAG 프레임워크의 전체 구조를 보여준다. 입력 질문은 하위 질문으로 분해된 뒤 GS-QG 모듈을 거쳐 선택적 질의로 확장되며, 검색된 문서를 기반으로 답변이 생성된다.

GS-QG는 두 단계로 구성된다. 첫째, 입력 질문을 의미 기반 그래프로 변환하여 기준점(anchor point)과 기준 구간(anchor section)을 식별한다. 둘째, 기준 구간 정보를 활용하여 핵심 질의와 보조 질의를 생성함으로써, 학습 기반 선택 모듈 없이도 중요 정보 중심의 선택적 검색을 수행한다.

1. Anchor-Aware Question Graph Construction

입력 질문을 의미 단위로 분해하여 개체 및 개념 간의 관계를 방향성 그래프로 표현한다. 이후 비교·조건 판단 등의 핵심 요소인 기준점(anchor point)과 동일 관계로 연결된 대상 노드 집합인 기준 구간(anchor section)을 식별한다. 이를 통해 질문의 핵심 구조를 보존하면서 선택적 검색이 가능하도록 한다.

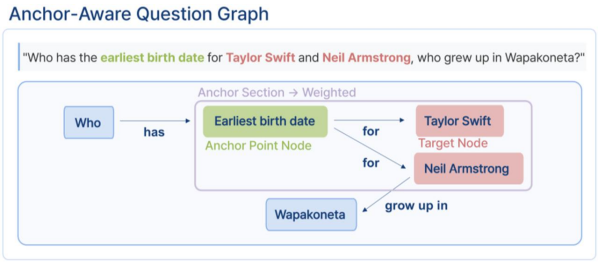

Fig. 2. Anchor-Aware Question Graph Construction Example

1.1 Structuring the Question Graph

입력 질문은 질문 분해 모듈을 통해 여러 하위 질문(sub-question)으로 분리되며, 각 하위 질문의 의미 단위를 추출하여 의미 기반 질문 그래프를 구성한다. 그래프는 개체(entity) 및 개념 단위를 노드로, 관계를 방향성 간선으로 표현한다.

Fig. 2는 이러한 질문 그래프 구성 결과를 예시로 나타낸 것이다. "Earliest birth date"가 기준점으로 식별되고, "Taylor Swift"와 "Neil Armstrong"이 비교 대상 노드로서 기준 구간을 형성한다. 동시에 "Wapakoneta"는 문맥 보조 노드로 연결되어 기준 구간을 보완한다.

본 연구의 질문 그래프 생성 과정은 DS-RAG의 구조를 그대로 계승한다. 구체적으로, (1) 각 하위 질문에서 주요 개체와 개념을 식별하여 노드로 변환하고, (2) 추출된 의미 단위 간의 관계를 방향성 간선으로 연결하여 하위 질문 단위의 부분 그래프를 형성한다. (3) 이후 각 부분 그래프를

통합하여 전체 질문의 의미적 맥락을 반영하는 단일 질문 그래프로 확장한다. 결과적으로 Fig. 2에 제시된 예시와 같이, 최종 질문 그래프는 입력 질문의 복잡한 의미 요소를 시각적으로 드러내며 이후 GS-QG 모듈의 기준점 및 기준 구간 식별 과정의 입력으로 활용된다.

1.2 Identifying Anchor Node and Anchor Section

완성된 질문 그래프를 기반으로 기준점과 기준 구간을 정의한다.

(1) 기준점 후보군 선정: 기준점 후보는 동일한 관계(relation)를 가지는 두 개 이상의 출력 간선을 지닌 노드로 선정한다. 이는 해당 노드가 여러 비교 대상이나 조건과 연결될 가능성을 시사한다.

(2) 기준점 노드 선정: 그래프의 루트 노드(root node)에 가장 가까운 후보 노드를 우선적으로 고려한다. 이는 일반적으로 질문의 상위 개념(비교 기준이나 조건 판단 등)에 해당하기 때문이다. 다만, 질문 구조가 상이한 경우에는 유연하게 적용할 수 있다.

(3) 기준 구간 정의: 기준점에서 동일 관계로 연결된 대상 노드들(target nodes)과 기준점을 포함한 집합을 기준 구간으로 정의한다.

Algorithm 1 Anchor Node and Anchor Section Identification

Require: G_q: Question graph where each node contains:
- `entity`: semantic element represented by the node
- `edges`: set of outgoing edges

Each edge contains:
- `relation`: semantic meaning of the connection

Ensure:
`anchor_node`: identified anchor node
`anchor_section`: group of target nodes and anchor node connected by the same relation

```
anchor_node ← NULL, anchor_section ← [ ]
component_list ← weakly_connected_components(G_q)
for all component G_c in component_list do
    G_c ← remove_cycles(G_c)
    root_nodes ← nodes in G_c with in-degree 0
    root_node ← root_nodes[0]
    current_node ← root_node
    while current_node is not a leaf do
        out_edges ← outgoing_edges(current_node)
        if has_duplicate_relation(out_edges) then
            anchor_node ← current_node
            anchor_section.add(current_node)
            targets ← 1-hop target nodes with duplicated relations
            anchor_section.add(targets)
            break
        end if
        current_node ← select_next_node(current_node)
    end while
end for
return anchor_node, anchor_section
```

Fig. 3. Algorithm of Anchor Node and Anchor Section Identification

Fig. 3은 기준점 및 기준 구간을 식별하는 알고리즘 절차를 나타낸다. 이 과정은 다음과 같은 절차로 수행된다. (i) 먼저 질문 그래프를 약 연결 컴포넌트(weakly connected component) 단위로 분리한다. (ii) 각 컴포넌트는 비순환 방향 그래프(Directed Acyclic Graph, DAG) 형태로 정제하여 불필요한 순환 관계를 제거한다. (iii) 이후 루트 노드를 식별하고, 여러 루트 노드가 존재할 경우 대표 루트 노드를 결정한다. (iv) 결정된 루트 노드로부터 하향 탐색을 수행하면서, 동일 관계로 분기되는 최초의 노드를 기준점으로 선정한다. (v) 마지막으로, 기준점에서 동일 관계로 연결된 대상 노드들을 포함하여 기준 구간을 정의한다. 이때, 루트 노드의 선정은 질문 내 등장 순서를 고려하여 가장 먼저 나타나는 개념을 우선하는 방식을 적용한다. 이러한 절차를 통해 질문 내 의미적 중심이 되는 노드를 체계적으로 식별할 수 있다.

2. Query Generation Based on Anchor Section

기준점-인식 질문 그래프를 기반으로, 기준점과 대상 노드 쌍을 중심으로 하위 질의 그래프(sub-query graph)를 구성한다. 이 하위 그래프는 기준점과 대상 노드의 관계뿐 아니라 주변 문맥을 포함함으로써, 보다 풍부한 의미적 검색 질의를 생성할 수 있다.

각 하위 질의 그래프는 두 가지 질의로 분리된다:

(1) 핵심 기준 질의(Core Anchor Query): 기준점과 하나의 대상 노드로 구성되며, 질문의 핵심 비교 조건이나 요구를 직접적으로 반영한다.

(2) 보조 문맥 질의(Supportive Context Query): 하위 질의 그래프에서 핵심 기준 질의를 제외한 주변 정보를 포함한다. 배경지식, 조건, 부가 설명 등 핵심 질의를 보완하는 정보를 제공하여, 검색 과정에서 의미적 손실을 최소화한다.

이 두 질의는 하나의 통합 질의 벡터로 결합되어 검색 단계에 전달된다. 결합 방식은 다음의 두 가지 방법으로 정의된다.

방법 1: 반복 질의 삽입(Query Repetition Anchoring): 핵심 기준 질의를 문서 임베딩 입력의 앞뒤에 반복 삽입하여, 내적 유사도 계산에서 해당 정보의 중요도를 강조한다. 이는 추가 연산 없이도 구조 기반 검색 성능을 향상시키는 간단하고 경량화된 접근이다.

방법 2: 선형 결합 기반 가중 질의(Weighted Dual-Query Fusion): 핵심 기준 질의 qc와 보조 문맥 질의 qs를 사전 학습된 인코더를 통해 각각 임베딩한 후, 정규화(normalization) 된 두 벡터를 선형 결합하여 최종 질의 벡터를 구성한다.

$$q_{final} = \alpha \cdot Norm(q_c) + (1-\alpha) \cdot Norm(q_s)$$

여기서 α는 핵심 내용과 문맥성 간의 상대적 중요도를 조

Table 1. Experimental Results on Ranking and Comparison Question Datasets

Data Type	Method	Question-to-Context Ratio	Precision	Recall	F1-Score	NS
Ranking Type	EPQD RAG		0.343	0.971	0.502	8.58
	DS-RAG		0.726	0.973	0.832	3.95
	Lightweight DS-RAG (Method 1)		0.710	0.742	0.719	3.18
	Lightweight DS-RAG (Method 2)	0.5:0.5	0.707	0.738	0.716	
		0.6:04	0.789	0.825	0.800	
		0.7:0.3	0.831	0.867	0.841	
		0.8:0.2	0.863	0.900	0.874	
		0.9:0.1	0.884	0.922	0.895	
		Average	0.815	0.850	0.825	
Comparison Type	EPQD RAG		0.353	0.850	0.494	7.48
	DS-RAG		0.921	0.925	0.923	3.01
	Lightweight DS-RAG (Method 1)		0.795	0.783	0.780	2.97
	Lightweight DS-RAG (Method 2)	0.5:0.5	0.726	0.714	0.712	
		0.6:0.4	0.856	0.838	0.838	
		0.7:0.3	0.904	0.884	0.884	
		0.8:0.2	0.912	0.895	0.893	
		0.9:0.1	0.911	0.894	0.892	
		Average	0.862	0.845	0.844	

절하는 하이퍼 파라미터이다. 이 방법은 검색 유형에 따라 적절한 α 값을 조정함으로써 질의의 민감도와 검색 집중도를 균형 있게 유지한다.

생성된 질의 쌍은 모두 동일한 질문 그래프를 기반으로 하므로, 질문 내의 주요 요소(비교 기준, 정렬 조건 등)가 의미 왜곡 없이 유지된다. 핵심 기준 질의는 기준점-대상 노드 쌍을 직접 포함하여 질문의 주요 비교 조건을 보존하고, 보조 문맥 질의는 그 주변 노드를 포함하여 추가적 배경 정보를 제공한다.

이러한 구조적 분리는 설계 단계에서 이미 질문의 의미 단위가 손실 없이 반영되도록 보장하며, 선택 모듈 없이도 질문 내 주요 비교 기준과 대상을 중심으로 검색을 수행한다. 또한 보조 질의를 통해 부가 정보를 함께 반영하므로, 정보 누락 없이 높은 검색 정밀도를 유지할 수 있다.

이 방식은 질문 그래프의 구조적 일관성에 기반하여 자동으로 질의 구성을 결정하기 때문에, 주관적 선택 개입이 최소화되고 일관된 질의 생성을 보장한다. 실험 결과에서 확인하듯, 이러한 질의 생성 구조는 기존 DS-RAG의 선택 모듈과 유사한 수준의 선택 성능을 더 경량화된 방식으로 구현한다.

3. Experiments and Results

본 연구는 GS-QG 기반의 선택적 질의 생성을 통해 DS-RAG 프레임워크를 경량화하면서도 선택 성능을 유지할 수 있는지를 검증하였다.

3.1 Dataset and Experimental Setup

본 연구에서는 DS-RAG[14]에서 정의한 Custom Multi-Document QA 데이터 셋을 사용하였다. 데이터 셋은 비교(Comparison) 및 순위(Ranking) 유형의 질문 각 500개씩, 총 1,000개의 질의로 구성되며, 각 질문은 최대 4개의 정답 청크(answer chunk)를 요구한다. 정답 청크는 여러 문서에 분산되어 존재하여, 기존 HotpotQA[21] 및 Multi-hop RAG[3]보다 정답 판단 명확성과 검색 정밀도 평가에 적합하도록 설계되었다.

문서 임베딩은 OpenAI의 text-embedding-ada-002 모델[22]을 사용하였고, 문서 청크는 길이 250, 오버랩 50으로 분할 후 FAISS[23] 인덱싱을 수행하였다. 검색 결과 비교의 공정성을 위해 모든 실험에서 Top-k=1로 고정하였다.

비교 대상은 다음 세 가지 방식이다. (i) EPQD RAG: 선택 모듈 없이 EPQD 기반으로만 검색을 수행한 RAG. (ii) DS-RAG: 선택 모듈(DICS)과 질문 분해를 모두 사용하는 기존 프레임워크. (iii) Lightweight DS-RAG: 제안한 GS-QG 기반 구조 (Method 1, Method 2). 이때 Method 1은 반복 질의 삽입 방식, Method 2는 선형 결합 기반 가중 질의 방식을 의미한다.

3.2 Evaluation Metrics

본 연구의 초점은 검색 단계의 정밀도와 불필요한 정보 최소화이다. 따라서 생성기 품질 평가는 제외하고, 검색 단계만을 정량적으로 평가하였다.

따라서 다음의 지표들을 중심으로 평가를 수행하였다:
(1) Precision / Recall / F1-score: 정답 포함률과 불필요한 문서 제거 능력을 측정한다.
(2) NS(Number of Selected Chunks): 선택된 문서 청크의 수를 의미하며, 정보 압축 효율성을 반영한다. 동일한

정답 청크가 중복 검색된 경우 1회만 계산한다.

3.3 Experimental Results and Analysis

Table 1은 랭킹 및 비교 유형에 대한 각 방법의 실험 결과를 나타낸다. Lightweight DS-RAG(Method 2)는 DS-RAG의 선택 모듈을 사용하지 않고도 평균 95% 수준의 F1-score를 유지하였다.

(1) EPQD vs Lightweight DS-RAG

EPQD는 모든 질의 정보를 균등하게 처리하기 때문에, 보조 정보가 과도하게 포함되어 Precision이 낮게 나타났다. 반면, Lightweight DS-RAG는 기준점 중심의 구조적 질의 생성을 통해 불필요한 정보 유입을 억제하고 높은 Recall을 유지하였다. 이는 GS-QG가 질문 내 비교 조건과 대상 정보를 구조적으로 반영한다는 객관적 근거가 된다. 그 결과 평균 F1-score가 50% 이상 향상, NS는 평균 3 이하로 감소하였다. 즉, 적은 수의 문서로도 정답 청크를 정확히 포함할 수 있는 구조적 효율성을 입증하였다.

(2) Lightweight DS-RAG vs DS-RAG

Method 2의 F1-score는 기존 DS-RAG 대비 약 95% 수준이며, Recall이 소폭 낮음에도 NS가 더 작아 효율성이 높았다. 이는 제안된 GS-QG 모듈이 학습 없이도 DS-RAG의 선택 모듈과 유사한 문서 선별 능력을 구조적으로 달성함을 보여준다.

(3) Method 1 vs Method 2

Method 1은 구현이 단순하고 경량성이 뛰어나지만, 질문 길이가 길어질수록 보조 정보가 핵심 정보를 희석시키는 경향이 있다. 반면, Method 2는 핵심·보조 질의를 분리 임베딩 후 선형 결합하여 다양한 질의 유형에서 안정적인 성능을 보였다. 따라서 확장성 및 강건성 측면에서 Method 2가 Method 1보다 우위를 가진다.

종합하면, 제안된 Lightweight DS-RAG는 (1) 선택 모듈이 없는 구조임에도 기존 DS-RAG의 약 95% 수준의 선택 성능을 유지하고, (2) 별도의 학습 과정 및 선택 모듈 호출이 불필요하여 추가 연산 자원이 요구되지 않으며, (3) EPQD 대비 평균 67.5%의 성능 향상을 달성하였다. 이는 질문 구조 기반 질의 생성만으로도 선택 효과를 구현할 수 있음을 실험적으로 입증한 결과이다. 결과적으로, 제안된 접근법은 도메인 학습 없이도 적용 가능한 실용적 대안으로, 특히 제한된 연산 자원 환경 기반의 상용 다중 문서 QA 시스템에 효과적으로 활용될 수 있음을 보여준다.

IV. Conclusions

본 연구는 자원이 제약된 환경에서도 별도의 학습 과정 없이 동작 가능한 Lightweight DS-RAG 프레임워크를 제안하였다. 이를 통해 질문 그래프 기반 선택 질의 생성(GS-QG) 모듈을 이용하여 핵심 요소를 구조적으로 반영하고, 선택 모듈 없이도 높은 검색 성능을 유지함을 실험적으로 입증하였다.

실험 결과, 제안된 프레임워크는 기존 DS-RAG 대비 약 95% 수준의 검색 성능을 유지하면서 연산 비용을 크게 절감하였다. 이는 복잡한 선택 모듈 없이도 구조적 설계만으로 정밀하고 효율적인 검색이 가능함을 보여준다.

향후 연구는 두 가지 방향으로 확장될 예정이다. 첫째, LLM 기반 질문 그래프 생성이 경량 환경에 부적합하다는 점을 고려하여, 소형 언어 모델(SLM) 기반의 그래프 생성 기법을 개발할 계획이다. 둘째, 명시적 기준점(anchor)에 의존하지 않고 기준점을 자동으로 추론하는 방식을 도입하여 시스템의 범용성을 강화할 예정이다.

ACKNOWLEDGEMENT

This work was supported by the 2025 Research Grant of Kwangwoon University.

REFERENCES

[1] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. Yih, T. Rocktäschel, S. Riedel, and D. Kiela, "Retrieval-augmented generation for knowledge-intensive NLP tasks," Advances in Neural Information Processing Systems, vol. 33, pp. 9459–9474, 2020. DOI: 10.48550/arXiv.2005.11401

[2] S. Barnett, S. Kurniawan, S. Thudumu, Z. Brannelly, and M. Abdelrazek, "Seven failure points when engineering a retrieval augmented generation system," in Proc. IEEE/ACM 3rd Int. Conf. AI Eng. - Softw. Eng. AI (CAIN), pp. 194–199, Lisbon, Portugal, Apr. 2024. DOI: 10.1145/3644815.3644945

[3] Y. Tang and Y. Yang, "MultiHop-RAG: Benchmarking Retrieval-Augmented Generation for Multi-Hop Queries," arXiv preprint, arXiv:2401.15391, Jan. 2024. DOI: 10.48550/arXiv.2401.15391

[4] X. Wu, S. Li, H.-T. Wu, Z. Tao, and Y. Fang, "Does RAG introduce unfairness in LLMs? Evaluating fairness in retrieval-augmented generation systems," in Proc. 31st Int. Conf.

on Computational Linguistics (COLING), pp. 10021–10036, Abu Dhabi, UAE, Jan. 2025. DOI: 10.48550/arXiv.2409.19804

[5] S. Gupta, R. Ranjan, and S. N. Singh, "A Comprehensive Survey of Retrieval-Augmented Generation (RAG): Evolution, Current Landscape and Future Directions," arXiv preprint arXiv:2410.12837, 2024. DOI: 10.48550/arXiv.2410.12837

[6] P. J. L. Ammann, J. Golde, and A. Akbik, "Question decomposition for retrieval-augmented generation," arXiv preprint, arXiv:2507.00355, Jul. 2025. DOI: 10.48550/arXiv.2507.00355

[7] E. Perez, P. Lewis, W. Yih, K. Cho, and D. Kiela, "Unsupervised question decomposition for question answering," in Proc. 2020 Conf. on Empirical Methods in Natural Language Processing (EMNLP), pp. 8864–8880, Online, Nov. 2020. DOI: 10.18653/v1/2020.emnlp-main.713

[8] X. Huang, S. Cheng, Y. Shu, Y. Bao, and Y. Qu, "Question decomposition tree for answering complex questions over knowledge bases," AAAI, vol. 37, no. 11, pp. 12924–12932, Jun. 2023. DOI: 10.1609/aaai.v37i11.26519

[9] K. Han and C. Gardent, "Generating complex question decompositions in the face of distribution shifts," in Proc. 2025 Conf. of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pp. 1189–1211, Albuquerque, New Mexico, Apr. 2025. DOI: 10.18653/v1/2025.naacl-long.55

[10] J. Zhang, S. Cao, T. Zhang, X. Lv, J. Li, L. Hou, J. Shi, and Q. Tian, "Reasoning over hierarchical question decomposition tree for explainable question answering," in Proc. 61st Annu. Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 14556–14570, Toronto, Canada, Jul. 2023. DOI: 10.18653/v1/2023.acl-long.814

[11] H. Zhang, J. Cai, J. Xu, and J. Wang, "Complex question decomposition for semantic parsing," in Proc. 57th Annu. Meeting of the Association for Computational Linguistics, pp. 4477–4486, Florence, Italy, Jul. 2019. DOI: 10.18653/v1/P19-1440

[12] M. Hasson and J. Berant, "Question decomposition with dependency graphs," arXiv preprint, arXiv:2104.08647, Apr. 2021. DOI: 10.48550/arXiv.2104.08647

[13] Z. Shi, W. Sun, S. Gao, P. Ren, Z. Chen, and Z. Ren, "Generate-then-Ground in Retrieval-Augmented Generation for Multi-hop Question Answering," arXiv preprint, arXiv:2406.14891, Sep. 2024. DOI: 10.48550/arXiv.2406.14891

[14] M. Kwon, J. Bang, S. Hwang, J. Jang, and W. Lee, "A Dynamic-Selection-Based, Retrieval-Augmented Generation Framework: Enhancing Multi-Document Question-Answering for Commercial Applications," Electronics, vol. 14, no. 4, p. 659, 2025. DOI: 10.3390/electronics14040659

[15] A. Ammar, A. Koubaa, O. Nacar, and W. Boulila, "Optimizing Retrieval-Augmented Generation: Analysis of Hyperparameter Impact on Performance and Efficiency," arXiv preprint, arXiv:2505.08445, May 2025. DOI: 10.48550/arXiv.2505.08445

[16] Y. Cao, J. Xu, T.-Y. Liu, H. Li, Y. Huang, and H.-W. Hon, "Adapting ranking SVM to document retrieval," in Proc. 29th Annu. Int. ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR '06), pp. 186–193, Seattle, WA, USA, 2006. DOI: 10.1145/1148170.1148205

[17] C. Pei, Y. Zhang, Y. Zhang, F. Sun, X. Lin, H. Sun, J. Wu, P. Jiang, J. Ge, W. Ou, and D. Pei, "Personalized re-ranking for recommendation," in Proc. 13th ACM Conf. on Recommender Systems (RecSys '19), pp. 3–11, Copenhagen, Denmark, 2019. DOI: 10.1145/3298689.3347000

[18] J. Gao, B. Chen, X. Zhao, W. Liu, X. Li, Y. Wang, W. Wang, H. Guo, and R. Tang, "LLM4Rerank: LLM-based Auto-Reranking Framework for Recommendations," in Proc. ACM on Web Conf. 2025 (WWW '25), pp. 228–239, Sydney, Australia, 2025. DOI: 10.1145/3696410.3714922

[19] R. Gangi Reddy, J. Doo, Y. Xu, M. A. Sultan, D. Swain, A. Sil, and H. Ji, "FIRST: Faster Improved Listwise Reranking with Single Token Decoding," in Proc. 2024 Conf. on Empirical Methods in Natural Language Processing (EMNLP), pp. 8642–8652, Miami, FL, USA, Nov. 2024. DOI: 10.18653/v1/2024.emnlp-main.491

[20] J. Lu, K. Hall, J. Ma, and J. Ni, "HYRR: Hybrid Infused Reranking for Passage Retrieval," in Proc. 2024 Joint Int. Conf. on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024), pp. 8528–8534, Torino, Italy, May 2024. DOI: 10.48550/arXiv.2212.10528

[21] Z. Yang, P. Qi, S. Zhang, Y. Bengio, W. Cohen, R. Salakhutdinov, and C. D. Manning, "HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering," in Proc. 2018 Conf. on Empirical Methods in Natural Language Processing (EMNLP), pp. 2369–2380, Brussels, Belgium, Oct.–Nov. 2018. DOI: 10.18653/v1/D18-1259

[22] OpenAI, “Vector embeddings,” OpenAI Platform Documentation, https://platform.openai.com/docs/guides/embeddings

[23] H. Jégou, M. Douze, and J. Johnson, "Faiss: A Library for Efficient Similarity Search," Engineering at Meta, https://engineering.fb.com/2017/03/29/data-infrastructure/faiss-a-library-for-efficient-similarity-search/

Authors

Ji-Min Bang received the B.S. degree in Computer Engineering from Kwangwoon University in 2025. Master's student Bang is currently pursuing her degree at the School of Computer and Information Engineering at Kwangwoon University. She is conducting research in a lab focused on information retrieval, agentic AI, and automated decision-making systems.

Woo-Sin Lee received the B.S., M.S. and Ph.D. degrees in Computer Engineering from Kwangwoon University, Korea, in 2001, 2003 and 2007, respectively. Dr. Lee was employed as a Software Engineer at Hanwha Systems Co., Ltd. from 2008 to 2022, and has been a faculty member of the School of Computer and Information Engineering at Kwangwoon University since 2023. His research interests focus on information retrieval, agentic AI, and automated decision-making systems.