외부 분류기를 활용한 반복적 검색 증강 생성의 효율성 개선

염시형°, 정근영, 이성희, 김학수
건국대학교 인공지능학과
{stv10121, jyjg7218, nlpshlee, nlpdrkim}@konkuk.ac.kr

Improving the Efficiency of Iterative Retrieval-Augmented Generation with an External Classifier

Sihyeong Yeom°, Geunyeong Jeong, Seonghee Lee, Harksoo Kim
Konkuk University, Department of Artifical Intelligence

요 약

검색 증강 생성은 질문과 관련된 정보를 검색하여 맥락으로 사용함으로써 대형 언어 모델이 학습할 때 보지 못한 질문에도 답변할 수 있게 만든다. 하지만, 일반적인 검색 증강 생성은 단 한 번의 검색과 생성을 수행하기 때문에 복잡한 질의를 효과적으로 다루는 데 한계가 존재한다. 최근 연구에서는 복잡한 질의를 하위 질의로 분해하고, 반복적으로 검색과 생성을 수행하는 학습 기반 반복적 검색 증강 생성이 뛰어난 성능을 보여주었지만, 언어 모델의 반복 호출로 인한 높은 계산 비용이 요구된다. 이에 본 연구는 외부 분류기를 활용하여 성능을 크게 저하시키지 않으면서도, 토큰 소비량을 효과적으로 줄일 수 있는 방법을 제안한다.

주제어: 검색 증강 생성(RAG), 다중홉 질의응답(Multi-hop Question Answering)

1. 서론

대형 언어 모델(Large Language Model, LLM)은 우수한 언어 이해 능력을 바탕으로 다양한 분야에서 뛰어난 성과를 보이고 있다[1, 2, 3]. 그러나 모델이 보유한 지식은 학습 시점에 고정되어 있어 학습 과정에서 접하지 못한 정보에 대해서는 부정확하거나 잘못된 답변을 생성할 수 있다[4]. 이는 언어 모델의 실제 활용에서 정확성과 신뢰성에 대한 우려를 불러일으키는 문제로, 이를 완화하기 위한 다양한 기법들이 제안되었다[5, 6, 7].

대표적으로 검색 증강 생성(Retrieval-Augmented Generation, RAG)이 있다[8, 9]. 검색 증강 생성은 위키피디아와 같은 고정된 지식 베이스나 웹으로부터 질문과 관련된 정보를 검색하고 문맥으로 제공함으로써, 모델이 보유하지 못한 최신 지식이나 학습 범위를 넘어서는 사실 기반 질문에도 보다 정확하고 신뢰성 있는 답변을 생성할 수 있도록 한다. 그러나 일반적인 검색 증강 생성 시스템은 단 한 번의 검색과 생성을 수행하기 때문에, 여러 번의 검색이 필요한 복잡한 질의에는 효과적으로 대응하기 어렵다.

최근 연구는 반복적인 검색과 생성을 결합한 반복적 검색 증강 생성 프레임워크를 통해 이러한 문제를 해결한다. Iter-RetGen[10], IRCoT[11]와 같은 프롬프트 기반 방법은 CoT(Chain-of-Thought) 또는 이에 준하는 형태의 추론 체인을 기반으로 단계별 검색과 생성을 반복함으로써, 다중홉 질의응답(Multi-hop Question Answering)에서 뚜렷한 성능 향상을 달성하였다. 다만 프롬프트에만 의존하는 방식은 일관성과 재현성이 부족하기 때문에 안정적인 성능 확보에 어려움이 있다. 이에 Auto-RAG[12], CoRAG(Chain-of Retrieval Augmented Generation)[13]와 같은 학습 기반 방법은 언어 모델이 반복적인 질의-검색-추론 형태의 체인을 생성하는 과정을 학습함으로써 프롬프트 기반 방법의 불안정성을 완화하고 보다 안정적이고 일관된 성능을 보여준다. 그러나 이러한 학습 기반 방법은 체인의 길이가 불필요하게 늘어날 경우 검색과 응답 생성을 반복하면서 상당한 토큰 비용이 발생한다는 한계를 지닌다.

본 연구에서는 외부 분류기를 통해 불필요한 반복을 최소화하면서도 복잡한 질의에 필요한 추론적 깊이를 유지하는 효율적인 반복적 검색 증강 생성 방안을 제안한다. 구체적으로, 각 단계에서 분류기를 활용하여 원본 질의를 해결하기에 충분한 근거가 확보되었는지를 판별함으로써, 모델이 불필요하게 체인을 끝까지 이어가지 않고도 조기에 답변을 도출할 수 있도록 설계하였다. 이를 통해 기존 학습 기반 방법의 불필요한 검색과 추론을 줄여 효율성을 이전보다 크게 향상시키고자 하였다. 실험 결과, 여러 다중홉 질의응답 벤치마크에서 샘플당 평균 토큰 소비량이 절반 가까이 줄었음에도, 성능이 기존 방법과 유사한 수준을 유지하거나 일부 데이터셋에서는 오히려 향상되는 결과를 확인함으로써 제안 방법의 효율성을 명확히 입증한다.

2. 관련 연구

일반적인 검색 증강 생성은 단 한 번의 검색 단계를 수행한 후 최종 답변을 생성하는 단일 단계 구조를 따른다. 그러나 이러한 단일 검색-생성 방식은 복잡한 다중 단계 추론을 요구하는 질문에 직면했을 때, 단일 검색만으로는 답변에 필요한 모든 정보를 한 번에 찾아내기 어렵다는 한계가 존재한다. 이를 보완하기 위해, 최근 연구들은 반복적인 검색과 생성을 결합한

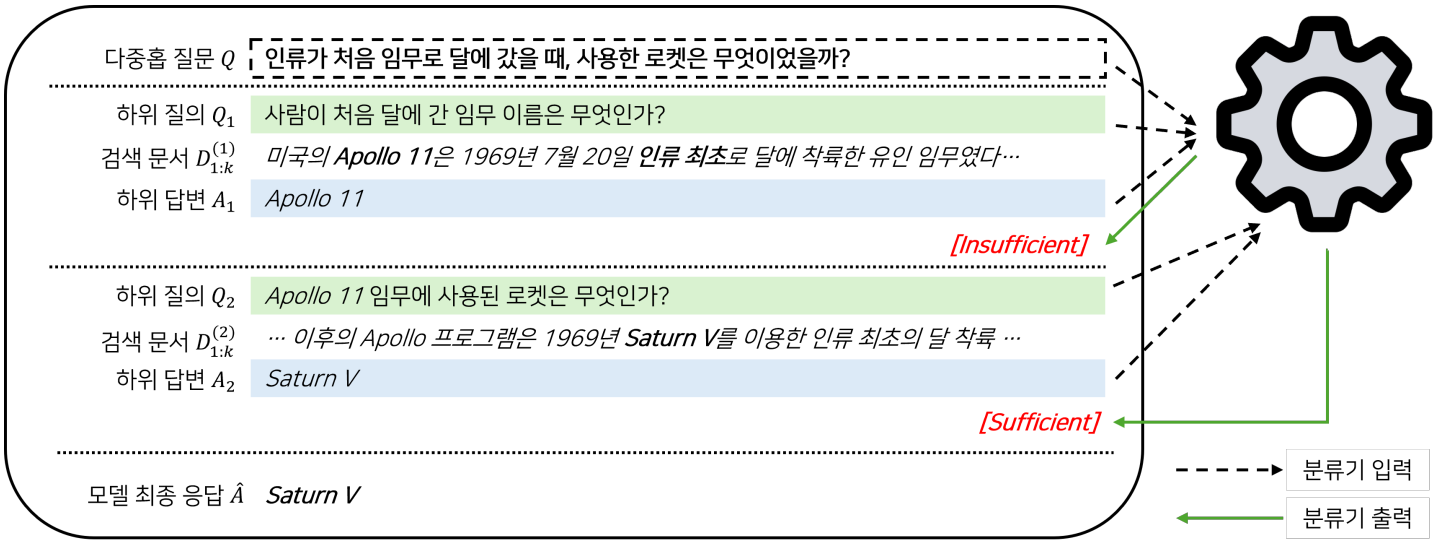

그림 1. 외부 분류기를 활용한 반복적 검색 증강 생성 효율성 개선 프레임워크

발전된 검색 증강 생성 시스템을 설계한다.

2.1 프롬프트 기반 반복적 검색 증강 생성

Iter-RetGen[10]은 질의로 정보를 검색한 뒤 응답을 생성하고, 생성된 응답을 다시 검색에 활용하는 과정을 반복함으로써 응답의 정확성을 높인다. 반면, IRCoT[11]는 CoT를 활용하여 생성과 검색을 번갈아 수행하며, 각 단계의 생성 결과로 다음 검색을 유도하고 추론을 보완한다. 두 방법 모두 단순한 프롬프트 설계만으로 구현이 가능하다는 장점이 있지만, 프롬프트 기반 접근만으로는 결과의 일관성과 안정성을 확보하기 어렵다는 한계가 존재한다.

2.2 학습 기반 반복적 검색 증강 생성

이러한 한계로 인해 최근에는 학습 기반 방법론들이 주목받고 있다. Auto-RAG[12]는 LLM이 자율적으로 추론하고 검색을 계획하여 반복적으로 정보를 수집함으로써, 기존 RAG보다 정확도와 유연성에서 큰 향상을 보인다. 반면, CoRAG[13]는 질의를 하위 질의로 분해하는 과정을 반복함으로써 여러 경로를 병렬적으로 탐색하도록 학습되어 다중홉 질의응답 과제에서 GPT-4o를 능가하는 우수한 성능을 보여준다.

그럼에도 불구하고, CoRAG는 추론 과정에서 사전에 정해진 체인 길이 혹은 일정한 횟수의 LLM 호출에 도달할 때까지 체인 생성을 반복하기 때문에, 이미 충분한 근거가 확보된 상황에서도 불필요한 단계를 모두 수행해야 하는 비효율적인 측면이 존재한다. 이러한 구조적 한계는 불필요한 탐색과 계산 자원 소모로 이어져 실제 응용에서 시스템의 효율성을 저해한다. 본 연구는 CoRAG를 기반으로 반복적 검색 증강 생성에서 효율적인 추론 경로 탐색을 위해, 외부 분류기를 활용한 동적 체인 길이 조절 기법을 새롭게 제안한다.

3. 제안 방법

3.1 합성 데이터셋 구성

외부 분류기를 학습하기 위해서는 각 단계에서 현재까지 확보된 근거가 충분한지를 판별할 수 있는 데이터가 필요하다. 이를 위해 본 연구에서는 멀티홉 QA 데이터셋에서 기각 샘플링 기법을 활용하여 합성 데이터셋을 구축하였다.

구체적으로, 하위 질의 Q_i는 원본 질의 Q와 그 이전의 하위 질의 집합 $Q_{<i}$ 및 하위 답변 집합 $A_{<i}$을 기반으로 하여 생성되며, 아래 수식 (1)과 같이 표현된다. 여기서 L은 사전에 정해진 최대 체인 길이를 의미한다.

$$Q_i = \mathrm{LLM}(Q_{<i}, A_{<i}, Q), \quad i \in [1, L] \tag{1}$$

이후, Q_i를 검색 질의로 사용하여 상위 k개의 관련 문서 $D_{1:k}^{(i)}$를 검색하고, 이를 바탕으로 하위 답변 A_i를 생성한다.

$$A_i = \mathrm{LLM}(Q_i, D_{1:k}^{(i)}), \quad i \in [1, L] \tag{2}$$

다음으로, 현재까지 누적된 $(Q, Q_{1:i}, A_{1:i})$와 Q로부터 검색된 문서 $D_{1:k}$를 기반으로, 모델은 예측 응답 $\hat{A}$를 생성한다.

$$\hat{A} = \mathrm{LLM}(Q, Q_{1:i}, A_{1:i}, D_{1:k}), \quad i \in [1, L] \tag{3}$$

만약, $\hat{A}$이 최종 정답 A와 일치한다면, `[Sufficient]` 토큰을 삽입하고 체인 생성을 종료한다. 반대로 $\hat{A}$과 A가 일치하지 않는다면 `[Insufficient]` 토큰을 삽입하고 체인 생성을 지속한다. 이러한 과정은 `[Sufficient]` 토큰이 삽입되거나 사전에 정해진 최대 길이 L에 도달할 때까지 반복된다. 만약 최대 체인 길이에 도달했음에도 A를 도출하지 못한 경우, 최종적으로 `[Insufficient]` 토큰을 삽입하고 체인 생성을 종료한다.

결국, 각 체인 C는 $(Q_{1:l}, A_{1:l})$의 튜플로 표현될 수 있으며, 여기서 $l(\leq L)$은 각 질의에 대해 실제로 생성된 체인의 길이를 의미한다.

고품질의 안정적인 학습 데이터셋을 확보하기 위해, 이와 같은 체인 생성 과정을 각 질의에 대해 n회 반복한다. 최종적으로 선택되는 체인 $\hat{C}$는 후보 체인 집합 $\mathcal{C} = \{C_1, C_2, \ldots, C_n\}$ 중에서 조건부 로그 우도가 최대인 체인으로 정의된다.

$$\hat{C} = \arg\max_{C \in \mathcal{C}} \log P(A \mid Q, C) \tag{4}$$

최종적으로, 분류기 학습을 위한 데이터셋 $\mathcal{T}$는 모든 질의 $Q^{(j)}$와 그에 대응하는 최적 체인 $\hat{C}^{(j)}$의 집합으로 구성된다.

$$\mathcal{T} = \{(Q^{(j)}, \hat{C}^{(j)}) \mid j = 1, \ldots, N\} \tag{5}$$

3.2 외부 분류기 학습

분류기는 원본 질의와 체인으로 구성된 시퀀스를 입력으로 받아, 해당 시점까지의 탐색 결과가 최종 답변 도출에 충분한지를 나타내는 이진 클래스 O_i를 출력하도록 학습된다. 학습은 교차 엔트로피 손실을 통해 수행되며, 각 단계 i에서의 손실은 아래 수식 (6)과 같다.

$$L_i = -\log \mathrm{P}(O_i \mid Q, \hat{C}), \quad i \in [1, L] \tag{6}$$

3.3 분류기를 활용한 동적 체인 길이 조절

그림 1과 알고리즘 1은 추론 과정에서 분류기를 활용하여 체인의 길이를 동적으로 조절하는 모습을 보여준다. 학습된 분류기는 추론 과정에서 각 단계마다 최종 응답 도출에 충분한 정보가 확보되었는지를 판별한다. 원본 질의 Q와 현재 단계 i까지의 하위 질의-답변 시퀀스 $(Q_{1:i}, A_{1:i})$가 입력으로 주어지면, 분류기는 `[Sufficient]` 또는 `[Insufficient]`을 출력한다. 분류기가 `[Sufficient]`을 출력하면, 언어 모델은 즉시 체인 생성을 종료하고 누적된 체인 정보를 바탕으로 최종 답변을 출력한다. 반대로 `[Insufficient]`가 출력되면, 다음 단계의 체인 생성이 지속된다. 이 과정은 분류기가 `[Sufficient]`을 출력하거나, 최대 체인 길이 L 혹은 사전에 정의된 LLM 호출 수에 도달할 때까지 반복된다.

이와 같은 절차를 통해, 본 연구는 CoRAG가 모든 단계를 끝까지 수행해야 했던 비효율성을 완화하고, 충분한 근거가 확보된 시점에서 조기 종료를 가능하게 하여 추론 효율성을 크게 향상시킨다.

Algorithm 1: 분류기를 활용한 체인 길이 동적 조절

Input: Original query Q, Maximum chain length L, Maximum LLM calls C, Retrieval doc size k
Output: Final answer $\hat{A}$
Initialize step $i \leftarrow 1$, call count $c \leftarrow 0$;
Initialize history $\mathcal{H} \leftarrow \emptyset$;
while $i \leq L$ ***and*** $c < C$ **do**
  Generate sub-question $Q_i \leftarrow \mathrm{LLM}(Q, \mathcal{H})$;
  Retrieve docs $D^{(i)}_{1:k} \leftarrow \mathrm{Retriever}(Q_i)$;
  Generate sub-answer $A_i \leftarrow \mathrm{LLM}(Q_i, D^{(i)}_{1:k})$;
  Update history $\mathcal{H} \leftarrow \mathcal{H} \cup \{(Q_i, A_i)\}$;
  $c \leftarrow c + 1$;
  Generate classifier output $O_i \leftarrow f(Q, \mathcal{H})$;
  if $O_i =$ *[Sufficient]* **then**
    Retrieve docs $D_{1:k} \leftarrow \mathrm{Retriever}(Q)$;
    Generate $\hat{A} \leftarrow \mathrm{LLM}(Q, \mathcal{H}, D_{1:k})$;
    break;
  $i \leftarrow i + 1$;
if $i > L$ ***or*** $c \geq C$ **then**
  Retrieve docs $D_{1:k} \leftarrow \mathrm{Retriever}(Q)$;
  Generate $\hat{A} \leftarrow \mathrm{LLM}(Q, \mathcal{H}, D_{1:k})$;
return $\hat{A}$

4. 실험 설정

4.1 데이터셋 및 평가 방법

분류기 학습을 위한 합성 데이터셋은 대표적인 멀티홉 질의응답 벤치마크인 HotpotQA[14], 2WikiMultihopQA[15], MuSiQue[16]의 학습 세트에서 각각 5,000개씩 무작위로 추출하여 구축하였다. 데이터 분포를 검토한 결과, 체인 길이가 1인 인스턴스가 과도하게 많았으며, 이는 충분한 정보가 축적되지 않은 초기 단계에서 분류기가 지나치게 조기 종료를 예측하도록 편향을 유발할 수 있다. 이를 방지하기 위해 체인 길이가 1인 데이터를 다운샘플링하여 결과적으로 10,000개의 훈련 인스턴스를 확보하였다. 체인 길이에 따른 데이터 분포는 표 1에서 확인할 수 있다.

표 1. 체인 길이에 따른 합성 데이터셋 분포

체인 길이	개수	비율(%)
1	4,077	40.8
2	2,325	23.2
3	871	8.7
4	481	4.8
5	351	3.5
6	1,895	19.0
총합	10,000	100

생성 모델의 성능 평가는 HotpotQA, 2WikiMultihopQA, MuSiQue의 검증 세트 전체와 Bamboogle[17]의 테스트 세트 전체를 활용하였으며, 총 22,523개의 질의를 대상으로 진행하였다. 각 평가 데이터셋의 분포는 표 2와 같다.

표 2. 평가 데이터셋 분포

데이터셋	개수	비율(%)
HotpotQA	7,405	32.9
2WikiMultihopQA	12,576	55.8
MuSiQue	2,417	10.7
Bamboogle	125	0.6
총합	22,523	100

평가 지표로는 Exact Match (EM), F1, 그리고 샘플당 평균 토큰 소비량(Average Token Consumption, ATC)을 사용하였다. 여기서 토큰 소비량은 검색 문서를 포함한 입력 토큰 수 T_{in}과 생성 토큰 수 T_{out}의 합이며, 샘플당 평균 토큰 소비량은 총 토큰 소비량을 전체 샘플 수 N으로 나눈 값으로 아래 수식 (7)과 같이 새롭게 정의하였다.

$$\mathrm{ATC} = \frac{1}{N}\sum_{j=1}^{N}\left(T_j^{\mathrm{in}} + T_j^{\mathrm{out}}\right) \tag{7}$$

또한, 문서 검색을 위한 지식 베이스로는 2018 위키피디아 덤프[18]를 사용하였으며, 총 35,678,075개의 문서로 구성된다.

4.2 모델

생성 모델로는 멀티홉 질의응답 데이터셋에 Llama-3.1-8B-Instruct[19]를 미세 조정한 **CoRAG-8B**[13]를 사용하였다. 해당 모델은 입력 질의로부터 하위 질의와 답변을 단계적으로 생성하도록 학습되어, 복잡한 질의를 점진적으로 분해하며 추론을 수행할 수 있다. 추론 시 디코딩 전략으로 본 연구에서는 greedy decoding을 적용하였다. 해당 디코딩 전략은 원본 질의로부터 하위 질의와 답변을 순차적으로 생성하며, 최대 체인 길이나 사전에 정의된 LLM 호출 수에 도달하면 최종 답변을 출력하는 방식이다.

또한, 본 연구에서는 CoRAG-8B를 기반으로 합성 데이터셋을 구축하고, 이를 바탕으로 외부 분류기를 학습하였다. 분류기로는 **FLAN-T5-large**[20]를 사용하였으며, 입력 질의와 누적된 하위 질의-답변 시퀀스를 기반으로 최종 답변 도출에 충분한 근거가 확보되었는지를 판별하도록 학습을 진행하였다. 분류기 학습을 위한 하이퍼 파라미터 설정은 표 3과 같다. 마지막으로, 검색 모델은 **BM25**[21]를 사용하였다. 원본 및 하위 질의로부터 검색하는 문서의 수는 5개로 설정하였다.

표 3. 분류기 학습을 위한 하이퍼 파라미터 설정값

하이퍼 파라미터	값
Peak Learning Rate	5×10^{-5}
Learning Rate Scheduler	Cosine
Learning Rate Warmup Ratio	0.05
Batch Size	8
Sequence Length	512
Optimizer	AdamW
Epochs	3

5. 실험 결과 및 분석

5.1 주요 결과

표 4는 제안한 방법의 주요 성능을 보여준다. $L = 6, \mathrm{greedy}$ 설정에 비해 분류기를 적용했을 때, 2WikiMultihopQA, HotpotQA, MusiQue에서는 EM과 F1이 소폭 감소했으나, 모든 벤치마크에서 샘플당 평균 토큰 소비량(ATC)이 크게 줄어들어 효율성이 뚜렷하게 개선되었다. 특히 Bamboogle에서는 ATC가 53.9% 수준으로 감소되었음에도 오히려 EM은 1.6%, F1은 0.4%가 향상되었다.

이는 분류기가 정답 도출에 충분한 근거가 확보된 시점을 정확히 판별하여 체인 생성을 조기에 종료함으로써 불필요한 탐색을 억제할 수 있음을 의미한다. 즉, 탐색 깊이를 무조건적으로 늘리기보다 분류기를 활용하여 적절한 시점에서 종료하는 것이 오히려 모델의 추론 안정성과 정답 도출 정확도를 높이는 데 중요함을 시사한다.

5.2 탐색 종료 시점의 적절성 평가

표 4의 $L = 1, \mathrm{greedy}$ 설정과 비교해봤을 때, 분류기를 적용한 경우에 평균 토큰 소비량이 증가하였다. 이는 분류기가 지나치게 조기 종료하지 않고, 탐색 과정에서 누적된 정보가 충분한지 여부를 적절히 판별했음을 의미한다. 다시 말해, 본 연구에서 제안한 방법은 단순히 탐색 단계만을 최소화하는 것이 아니라, 정답 도출에 필요한 근거가 충분히 확보된 시점에서만 종료함으로써 효율성과 성능적 강점을 동시에 확보할 수 있음을 보여준다.

5.3 분류기 크기에 따른 성능 차이 분석

분류기의 크기에 따른 성능 차이를 측정하기 위해 추가 실험을 수행하였다. 표 5의 결과에서 보듯, 분류기의 파라미터 수가 증가할수록 정확도가 꾸준히 향상되었으며, 이는 더 큰 모델이 하위 질의-답변 시퀀스를 기반으로 정보 충분성 여부를 보다 정밀하게 판별할 수 있음을 시사한다. 모델 크기가 커질수록 계산

	2WikiQA			HotpotQA			Bamboogle			MuSiQue		
	EM	F1	ATC	EM	F1	ATC	EM	F1	ATC	EM	F1	ATC
CoRAG-8B												
▷ $L = 1$, greedy	40.2	46.3	1,110	35.3	47.4	1,094	31.2	42.8	1,070	11.6	21.2	1,090
▷ $L = 6$, greedy	**58.8**	**63.9**	7,868	**44.8**	**57.8**	7,684	33.6	45.9	8,218	**18.6**	**29.1**	7,950
+ ***classifier***	56.2	61.2	3,455	42.0	54.7	3,809	**35.2**	**46.3**	4,436	18.2	28.1	5,836
Δ	↓ 2.6%	↓ 2.7%	↓ **56.1%**	↓ 2.8%	↓ 3.1%	↓ **50.4%**	↑ **1.6%**	↑ **0.4%**	↓ **46.0%**	↓ 0.4%	↓ 1.0%	↓ **26.6%**

표 4. 멀티홉 질의응답 데이터셋에서의 성능 결과표. Δ는 $L = 6$, greedy를 기준으로 한 **+*classifier*** 조건의 성능 변화량을 의미한다.

비용과 메모리 사용량이 증가하지만, 본 연구에서는 분류기의 판별 정확도가 전체 시스템 효율성에 직접적인 영향을 미칠 것을 고려하여 FLAN-T5-large를 최종 분류기로 채택하였다. 이를 통해 분류기의 예측 정확도를 극대화하고, 체인 생성의 조기 종료 여부를 한층 안정적으로 판별할 수 있도록 하였다.

표 5. 분류기 크기에 따른 성능 차이

Models	**Accuracy**
FLAN-T5-small (80M)	0.7302
FLAN-T5-base (250M)	0.7556
FLAN-T5-large (780M)	**0.8041**

6. 결론

본 연구에서는 CoRAG에 외부 분류기를 결합하여 추론 과정에서 체인 길이를 동적으로 조절하는 방법을 제안하였다. 이를 통해 다중홉 질의응답 벤치마크에서 기존 CoRAG가 성능적 강점을 상당 부분 유지하면서도 불필요한 탐색을 줄여 계산 비용을 크게 절감하였다. 특히, Bamboogle 벤치마크에서는 토큰 소비량을 감소시키는 동시에 응답 정확도가 향상되어 제안 방법의 효율성과 효과성을 모두 입증하였다.

그러나 본 연구는 자원 제약으로 인해 밀집 검색 모델을 사용하지 못했으며, 다양한 디코딩 전략을 적용하지 못하였다. 향후 연구에서는 보다 다양한 검색 모델 및 디코딩 전략을 활용하고, 분류기와 언어 모델의 상호작용을 정교하게 최적화함으로써 효율성과 성능을 동시에 극대화할 수 있을 것으로 기대된다.

감사의 글

이 성과는 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (RS-2025-00553041, 신뢰가능 대화 에이전트 구현을 위한 대형언어모델의 이성적 감성적 지능 강화).

참고문헌

[1] S. Minaee, T. Mikolov, N. Nikzad, M. Chenaghlu, R. Socher, X. Amatriain, and J. Gao, "Large language models: A survey," 2025. [Online]. Available: https://arxiv.org/abs/2402.06196

[2] OpenAI, "Gpt-4o system card," 2024. [Online]. Available: https://arxiv.org/abs/2410.21276

[3] DeepSeek-AI, "Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning," 2025. [Online]. Available: https://arxiv.org/abs/2501.12948

[4] L. Huang, W. Yu, W. Ma, W. Zhong, Z. Feng, H. Wang, Q. Chen, W. Peng, X. Feng, B. Qin, and T. Liu, "A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions," *ACM Transactions on Information Systems*, Vol. 43, No. 2, p. 1–55, Jan. 2025. [Online]. Available: http://dx.doi.org/10.1145/3703155

[5] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," 2021. [Online]. Available: https://arxiv.org/abs/2106.09685

[6] N. Zhang, Y. Yao, B. Tian, P. Wang, S. Deng, M. Wang, Z. Xi, S. Mao, J. Zhang, Y. Ni, S. Cheng, Z. Xu, X. Xu, J.-C. Gu, Y. Jiang, P. Xie, F. Huang, L. Liang, Z. Zhang, X. Zhu, J. Zhou, and H. Chen, "A comprehensive study of knowledge editing for large language models," 2024. [Online]. Available: https://arxiv.org/abs/2401.01286

[7] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. tau Yih, T. Rocktäschel, S. Riedel, and D. Kiela, "Retrieval-augmented generation for knowledge-intensive nlp

tasks," 2021. [Online]. Available: https://arxiv.org/abs/2005.11401

[8] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, M. Wang, and H. Wang, "Retrieval-augmented generation for large language models: A survey," 2024. [Online]. Available: https://arxiv.org/abs/2312.10997

[9] O. Ram, Y. Levine, I. Dalmedigos, D. Muhlgay, A. Shashua, K. Leyton-Brown, and Y. Shoham, "In-context retrieval-augmented language models," 2023. [Online]. Available: https://arxiv.org/abs/2302.00083

[10] Z. Shao, Y. Gong, Y. Shen, M. Huang, N. Duan, and W. Chen, "Enhancing retrieval-augmented large language models with iterative retrieval-generation synergy," 2023. [Online]. Available: https://arxiv.org/abs/2305.15294

[11] H. Trivedi, N. Balasubramanian, T. Khot, and A. Sabharwal, "Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions," 2023. [Online]. Available: https://arxiv.org/abs/2212.10509

[12] T. Yu, S. Zhang, and Y. Feng, "Auto-rag: Autonomous retrieval-augmented generation for large language models," 2024. [Online]. Available: https://arxiv.org/abs/2411.19443

[13] L. Wang, H. Chen, N. Yang, X. Huang, Z. Dou, and F. Wei, "Chain-of-retrieval augmented generation," 2025. [Online]. Available: https://arxiv.org/abs/2501.14342

[14] Z. Yang, P. Qi, S. Zhang, Y. Bengio, W. W. Cohen, R. Salakhutdinov, and C. D. Manning, "Hotpotqa: A dataset for diverse, explainable multi-hop question answering," 2018. [Online]. Available: https://arxiv.org/abs/1809.09600

[15] X. Ho, A.-K. D. Nguyen, S. Sugawara, and A. Aizawa, "Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps," 2020. [Online]. Available: https://arxiv.org/abs/2011.01060

[16] H. Trivedi, N. Balasubramanian, T. Khot, and A. Sabharwal, "Musique: Multihop questions via single-hop question composition," 2022. [Online]. Available: https://arxiv.org/abs/2108.00573

[17] O. Press, M. Zhang, S. Min, L. Schmidt, N. A. Smith, and M. Lewis, "Measuring and narrowing the compositionality gap in language models," 2023. [Online]. Available: https://arxiv.org/abs/2210.03350

[18] V. Karpukhin, B. Oğuz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and W. tau Yih, "Dense passage retrieval for open-domain question answering," 2020. [Online]. Available: https://arxiv.org/abs/2004.04906

[19] A. Grattafiori, A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Vaughan, A. Yang, A. Fan, and A. G. et al., "The llama 3 herd of models," 2024. [Online]. Available: https://arxiv.org/abs/2407.21783

[20] H. W. Chung, L. Hou, S. Longpre, B. Zoph, Y. Tay, W. Fedus, Y. Li, X. Wang, M. Dehghani, S. Brahma, A. Webson, S. S. Gu, Z. Dai, M. Suzgun, X. Chen, A. Chowdhery, A. Castro-Ros, M. Pellat, K. Robinson, D. Valter, S. Narang, G. Mishra, A. Yu, V. Zhao, Y. Huang, A. Dai, H. Yu, S. Petrov, E. H. Chi, J. Dean, J. Devlin, A. Roberts, D. Zhou, Q. V. Le, and J. Wei, "Scaling instruction-finetuned language models," 2022. [Online]. Available: https://arxiv.org/abs/2210.11416

[21] S. Robertson, S. Walker, M. Hancock-beaulieu, M. Gatford, and A. Payne, "Okapi at trec4," 01 1995.