RAPTOR: Recursive Abstractive Processing for Tree-Organized Retrieval

Announcement Date	@2025년 1월 16일
Conference Name	ICLR 2024
Keywords	LLM RAG

Abstract

- Retrieval-Based Language Model 은 세계 상태 변화에 적응하고 희소한 지식을 효과적으로 통합할 수 있지만, 기존 방법은 문서의 전체 맥락을 이해하는 데 한계가 있음
- RAPTOR 모델은 텍스트 조각을 재귀적으로 요약하여 하위부터 상위까지 요약 수준이 다른 트리를 생성하고, 이를 통해 긴 문서의 정보를 다양한 추상화 수준에서 통합
- 재귀적 요약을 활용한 리트리벌은 기존 방법보다 여러 작업에서 성능을 크게 향상시킴
- 복잡한 질문-응답 작업에서 RAPTOR와 GPT-4의 결합은 QuALITY 벤치마크에서 기존 최고 성능을 20% 절대 정확도 기준으로 개선

Introduction

- 대규모 언어 모델(LLM)은 매개변수에 인코딩된 지식으로 효과적인 지식 저장소 역할을 하지만, 특정 도메인 지식을 충분히 포함하지 못하며 세상의 변화로 인해 사실이 무효화될 수 있음
- 이를 해결하기 위해 Retrieval-Augmented 방식이 제안되었지만, 기존 방식은 짧고 연속적인 텍스트만 검색하여 대규모 담론 구조를 잘 나타내지 못하는 한계가 존재
- RAPTOR 시스템은 텍스트 조각을 클러스터링하고 요약하여 트리를 생성하며, 다양한 추상화 수준의 문맥을 제공해 이러한 문제를 해결함
- 이 트리 구조를 통해 RAPTOR는 Long Context 의 다양한 수준에서 정보를 통합하여 효과적이고 효율적으로 질문에 답변할 수 있음

- UnifiedQA, GPT-3, GPT-4와의 실험 결과, RAPTOR는 기존 리트리벌 증강 방식을 능가했으며 NarrativeQA, QASPER, QuALITY 데이터셋에서 새로운 최첨단 성과를 달성
- 특히 RAPTOR는 긴 문서를 분석하고 통합하여 복잡한 질문에도 높은 정확도를 제공하며, 현재까지의 접근법 중 가장 효율적인 방법으로 입증됨

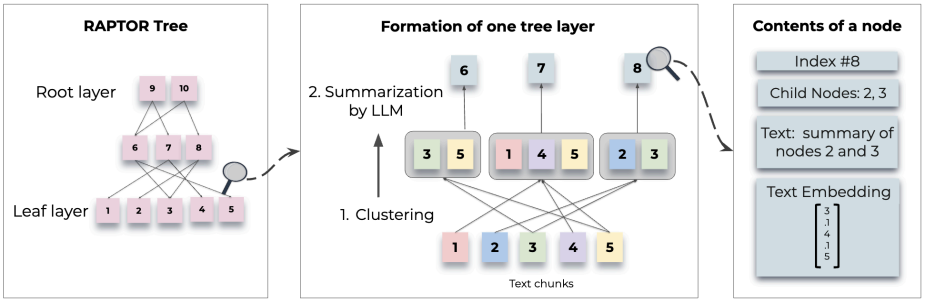

Figure 1: **Tree construction process:** RAPTOR recursively clusters chunks of text based on their vector embeddings and generates text summaries of those clusters, constructing a tree from the bottom up. Nodes clustered together are siblings; a parent node contains the text summary of that cluster.

Methods

- Overview of RAPTOR
  - RAPTOR는 Long Text가 종종 하위 주제와 계층적 구조를 가진다는 기존의 아이디어에 기반하여, 읽기에서 의미적 깊이와 연결성을 해결하기 위해 설계됨
  - 이는 더 넓은 주제적 이해와 세부 사항 간의 균형을 맞추며, 텍스트의 순서 뿐만 아니라 의미적 유사성에 따라 노드를 그룹화할 수 있는 재귀적 트리 구조를 만듦

  - RAPTOR 트리의 구축은 Retrieval Corpus를 전통적인 retrieval-augmented 기술과 유사하게 길이 100의 짧고 연속적인 텍스트로 나누는 것으로 시작
  - 문장이 100토큰을 초과할 경우, 문장을 중간에 자르지 않고 전체 문장을 다음 조각으로 옮겨 문맥적 및 의미적 일관성을 유지 (조각=chunk)
  - 이후 이 텍스트들은 SBERT라는 BERT기반 인코더를 사용해 임베딩되며, 조각과 그 임베딩은 트리 구조의 리프 노드를 형성

  - 비슷한 텍스트 조각을 그룹화하기 위해 클러스터링 알고리즘을 사용

    - 클러스터링된 텍스트는 LLM을 사용해 요약되며, 이러한 요약 텍스트는 다시 임베딩되고, **임베딩→클러스터링→요약**의 과정이 추가적인 클러스터링이 불가능해질 때까지 반복되어 원본 문서의 구조화된 다층 트리가 생성됨

    - 트리 내 질의를 위해 두 가지 전략을 도입함 : 트리 순회(tree traversal) 과 축약 트리(collapsed tree)
    - 트리 순회 방식은 트리를 계층별로 순회하여 각 수준에서 가장 관련성이 높은 노드를 선택
    - 축약 트리 방식은 모든 계층에 걸쳐 노드를 집합적으로 평가해 가장 관련성이 높은 노드를 찾음

- Clustering Algorithm
    - RAPTOR는 클러스터링을 통해 Text Segment 를 유의미한 그룹으로 조직화하며, Soft Clustering 방식을 사용해 텍스트가 여러 클러스터에 속할 수 있도록 유연성을 제공
    - 클러스터링에는 가우시안 혼합 모델(Gaussian Mixture Models, GMM) 을 사용하며, 고차원 데이터 문제를 해결하기 위해 UMAP 으로 차원을 축소하고 계층적 클러스터링을 수행
    - 전역 클러스터를 먼저 식별한 후, 지역 클러스터링을 추가로 적용하여 텍스트 데이터의 광범위한 관계를 포착
    - 최적 클러스터 개수는 BIC(Bayesian Information Criterion)을 통해 결정하며, 기대-최대화 알고리즘(Expectation-Maximization Algorithm)으로 GMM 매개변수를 추정
- Model-Based Summarization
    - GMM을 통해 클러스터링된 노드는 gpt-3.5-turbo 언어 모델을 통해 요약되어 large chunks를 간결하고 일관된 요약으로 변환
    - 요약 과정은 대량의 검색 정보를 압축하여 관리 가능한 크기로 만들며, 약 4%의 요약에서 경미한 환각이 발견되었으나 이는 상위 노드나 작업 결과에 영향을 미치지 않았음
- Querying
    - 이 섹션에서는 RAPTOR가 사용하는 두 가지 Querying 메커니즘인 트리 순회(tree traversal)와 축약 트리(collapsed tree)에 대해 설명

- 이 방법들은 다계층 RAPTOR 트리를 탐색하여 관련 정보를 검색하는 고유한 방식을 제공하며, 각각 고유의 장점과 단점이 존재

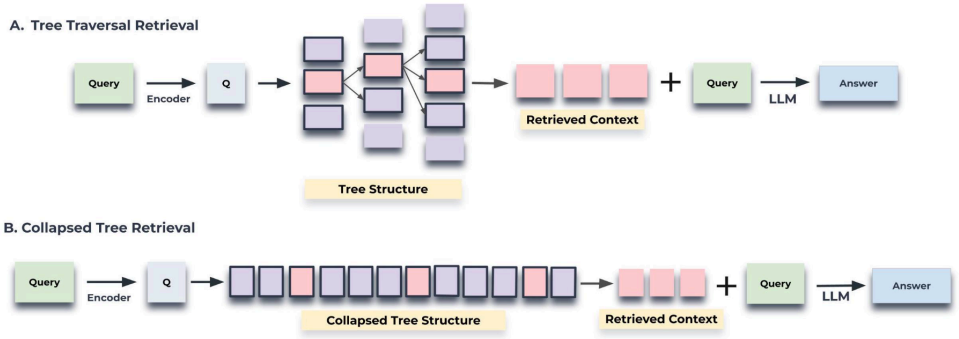

Figure 2: **Illustration of the tree traversal and collapsed tree retrieval mechanisms.** Tree traversal starts at the root level of the tree and retrieves the top-k (here, top-1) node(s) based on cosine similarity to the query vector. At each level, it retrieves the top-k node(s) from the child nodes of the previous layer's top-k. Collapsed tree collapses the tree into a single layer and retrieves nodes until a threshold number of tokens is reached, based on cosine similarity to the query vector. The nodes on which cosine similarity search is performed are highlighted in both illustrations.

- 트리 순회 (Tree Traversal) 방식
  - 트리 순회 방식은 루트 노드에서 시작해 질의 임베딩과 노드 임베딩 간 코사인 유사도를 계산해 상위 k개의 관련 노드를 선택
  - 선택된 노드의 자식 노드에서 다시 코사인 유사도를 계산하고 상위 k개의 노드를 선택하는 과정을 반복해 리프 노드까지 탐색
  - 탐색 과정에서 각 계층의 노드 집합(S1, S2, ..., Sd)을 생성하며, 최종적으로 이 집합들을 연결하여 관련 문맥을 조합
  - 탐색 깊이(d)와 각 계층에서 선택된 노드 수(k)를 조정함으로써 검색 정보의 구체성과 범위를 제어할 수 있음
  - 이 방법은 상위 계층에서 시작해 넓은 주제를 탐색하고, 하위 계층으로 갈수록 세부 정보에 점차 집중하는 방식을 제공
  - RAPTOR의 트리 순회 알고리즘은 다층 구조의 텍스트를 효과적으로 탐색하여 질의에 적합한 정보를 구성
- 축약 트리(collapsed tree) 방식
  - 축약 트리 방식은, 트리의 모든 노드를 단일 계층으로 평평하게 만들어 코사인 유사도를 기반으로 상위 k개의 노드를 선택하며, 검색을 간단하고 유연하게 만듬

- 트리 순회 방식과 비교해 축약 트리는 질문에 따라 적절한 수준의 정보를 효과적으로 검색해 QASPER 데이터셋에서 더 나은 성능을 보임 (Figure 3)

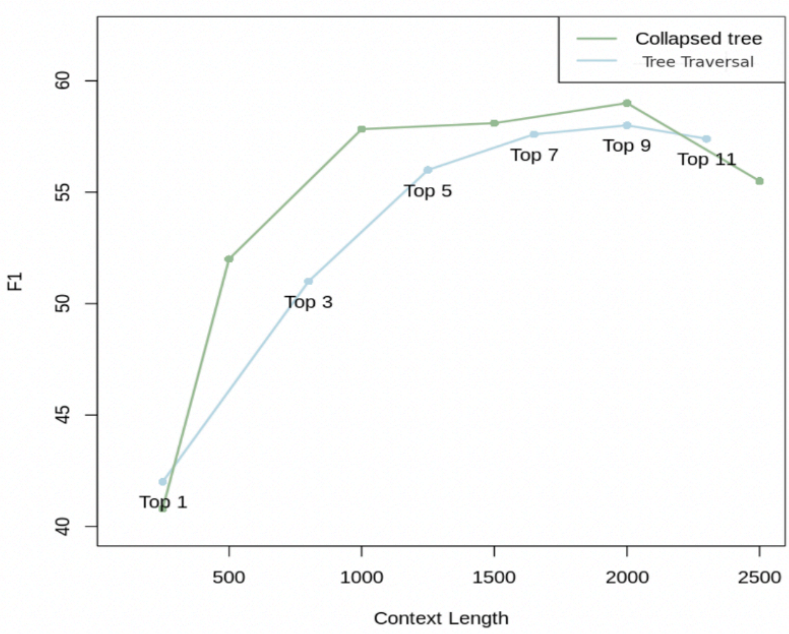

Figure 3: **Comparison of querying methods.** Results on 20 stories from the QASPER dataset using tree traversal with different top-k values, and collapsed tree with different context lengths. Collapsed tree with 2000 tokens produces the best results, so we use this querying strategy for our main results.

질의 방법 비교. QASPER 데이터셋의 20개 스토리에 대해 테스트. 2000토큰을 사용하는 축약 트리 방식이 가장 좋은 결과를 보여 해당 질의 전략을 채택.

- 이 접근법은 모든 노드에 대해 코사인 유사도 검색을 수행해야 한다는 단점이 있지만, FAISS와 같은 빠른 알고리즘으로 효율성을 높일 수 있음
- 따라서, 축약 트리 방식은 높은 유연성과 성능 때문에 RAPTOR의 주요 질의 메커니즘으로 채택됨

- Qualitative Study
  - 저자는 RAPTOR의 retrieval 과정이 DPR(Dense Passage Retrieval) 방식에 비해 가지는 이점을 이해하기 위해 정성적 분석을 수행

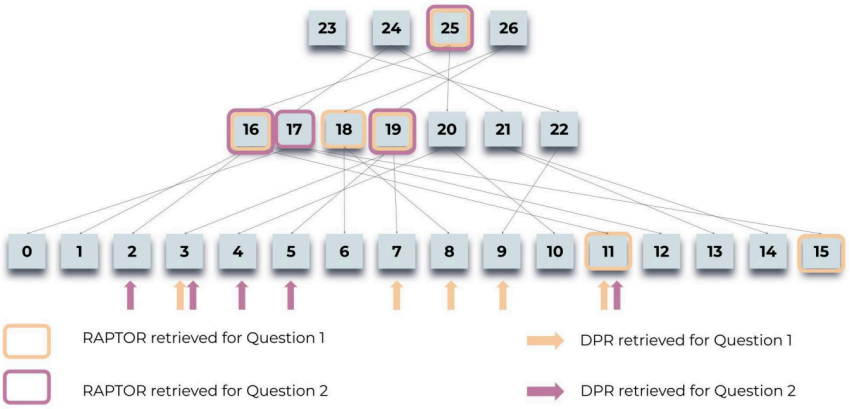

Figure 4: **Querying Process:** Illustration of how RAPTOR retrieves information for two questions about the Cinderella story: "What is the central theme of the story?" and "How did Cinderella find a happy ending?". Highlighted nodes indicate RAPTOR's selections, while arrows point to DPR's leaf nodes. Notably, RAPTOR's context often encompasses the information retrieved by DPR, either directly or within higher-layer summaries.

- 1500단어 분량의 신데렐라 동화를 활용해 주제적이고(thematic), 멀티홉 질문 분석했으며, RAPTOR가 downstream task에서 우수한 성과를 보였음

Experiments

- Datasets
  - RAPTOR의 성능은 NarrativeQA, QASPER, QuALITY라는 세 가지 질문-응답 데이터셋을 통해 평가
  - NarrativeQA는 책과 영화 대본의 전체 텍스트를 기반으로 하며, 긴 문학 텍스트를 이해하는 모델의 능력을 테스트
  - QASPER는 1,585개의 NLP 논문에서 추출된 5,049개의 질문으로 구성되며, 다양한 유형의 답변에 대해 F1 점수로 정확도를 측정
  - QuALITY는 약 5,000 토큰의 문맥과 다중 선택 질문으로 이루어져 있으며, QuALITY-HARD 하위 집합을 포함해 중간 길이 문서에서의 retrieval 성능을 평가
- Controlled Baseline Comparisons
  - UnifiedQA 3B를 reader 로 설정하고, SBERT/BM25/DPR 을 포함하거나 포함하지 않은 RAPTOR 트리 구조를 사용하는 임베딩 모델과 함께 세 가지 데이터셋에서 통제된 비교를 제시

Table 1: **NarrativeQA Performance With + Without RAPTOR:** Performance comparison of various retrieval methods (SBERT, BM25, DPR) with and without RAPTOR on the NarrativeQA dataset, using UnifiedQA-3B as the language model. RAPTOR outperforms baselines of each respective retrieval method.

Model	ROUGE	BLEU-1	BLEU-4	METEOR
SBERT with RAPTOR	**30.87%**	**23.50%**	**6.42%**	**19.20%**
SBERT without RAPTOR	29.26%	22.56%	5.95%	18.15%
BM25 with RAPTOR	**27.93%**	**21.17%**	**5.70%**	**17.03%**
BM25 without RAPTOR	23.52%	17.73%	4.65%	13.98%
DPR with RAPTOR	**30.94%**	**23.51%**	**6.45%**	**19.05%**
DPR without RAPTOR	29.56%	22.84%	6.12%	18.44%

Table 2: **QuALITY and QASPER Performance With + Without RAPTOR:** Performance comparison across the QuALITY and QASPER datasets of various retrieval methods (SBERT, BM25, DPR) with and without RAPTOR. UnifiedQA-3B is used as the language model. RAPTOR outperforms baselines of each respective retrieval method for both datasets.

Model	Accuracy (QuALITY)	Answer F1 (QASPER)
SBERT with RAPTOR	**56.6%**	**36.70%**
SBERT without RAPTOR	54.9%	36.23%
BM25 with RAPTOR	**52.1%**	**27.00%**
BM25 without RAPTOR	49.9%	26.47%
DPR with RAPTOR	**54.7%**	**32.23%**
DPR without RAPTOR	53.1%	31.70%

- 표 1과 2에서, 저자는 RAPTOR가 어떤 검색기와 결합하더라도 모든 데이터셋에서 기존보다 높은 성능을 보여줬음을 제시
- SBERT를 사용하는 RAPTOR 가 가장 높은 성능을 보였기 때문에, 이후 모든 실험에서 이를 사용

Table 3: Controlled comparison of F-1 scores on the QASPER dataset, using three different language models (GPT-3, GPT-4, UnifiedQA 3B) and various retrieval methods. The column ”Title + Abstract” reflects performance when only the title and abstract of the papers are used for context. RAPTOR outperforms the established baselines BM25 and DPR across all tested language models. Specifically, RAPTOR’s F-1 scores are at least 1.8% points higher than DPR and at least 5.3% points higher than BM25.

Retriever	GPT-3 F-1 Match	GPT-4 F-1 Match	UnifiedQA F-1 Match
Title + Abstract	25.2	22.2	17.5
BM25	46.6	50.2	26.4
DPR	51.3	53.0	32.1
RAPTOR	**53.1**	**55.7**	**36.6**

- 표 3에서는 QASPER 데이터셋에 대한 F1 Score 인데, 해당 데이터셋은 NLP 논문 내의 정보를 종합하는 작업이 필요하기 때문에 RAPTOR의 상위 수준 요약 노드가

단순히 가장 유사한 텍스트 청크만 추출할 수 있는 다른 방법들보다 더 나은 결과를 보이는 것은 놀라운 것이 아님

- 기존의 방법들은 개별 청크만으로 올바른 응답을 포함하지 못함