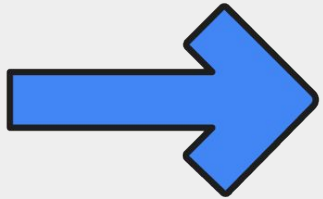
Google Developer Group
Incheon National University

Retrieval-Augmented Generation**(RAG)** for Knowledge-Intensive NLP Tasks



GDGoC AI Seminar **Member** 홍은진

기존 언어모델의 한계

Parametric 모델?

모델 내부 파라미터에 저장

ex) GPT, BERT, BART

같은 언어모델은 학습할 때 파라미터(가중치)에 세상의 지식을 저장

그들의 한계

- 정보를 수정하거나 확장하기 어렵고
- 왜 그렇게 예측했는지 쉽게 설명하지 못하며
- 사실이 아닌 내용을 만들어낼 수 있다.

파라미터 기반 메모리 + 검색 기반 메모리와 결합

Retrieval-Augmented Generation**(RAG)**

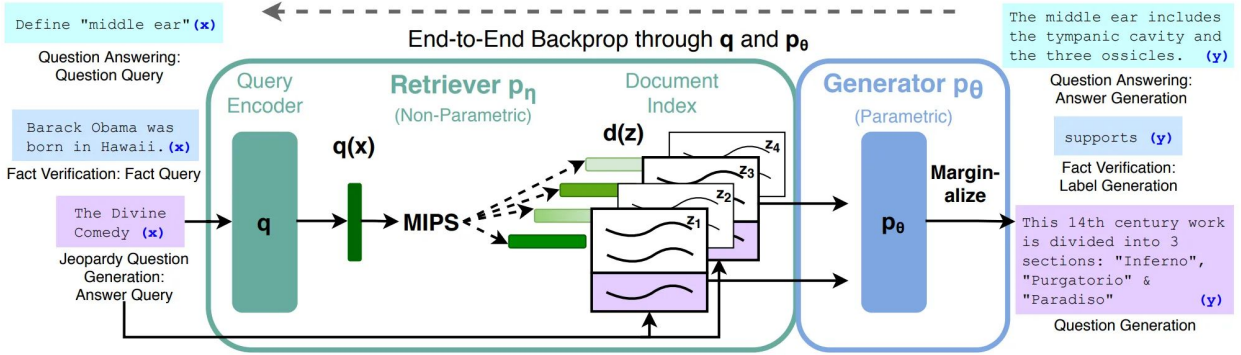


Figure 1: Overview of our approach. We combine a pre-trained retriever (*Query Encoder* + *Document Index*) with a pre-trained seq2seq model (*Generator*) and fine-tune end-to-end. For query x, we use Maximum Inner Product Search (MIPS) to find the top-K documents z_i. For final prediction y, we treat z as a latent variable and marginalize over seq2seq predictions given different documents.

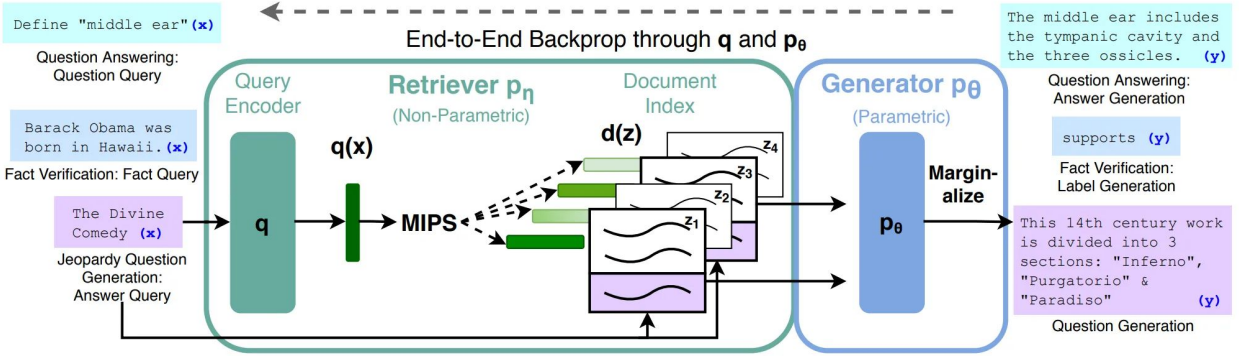
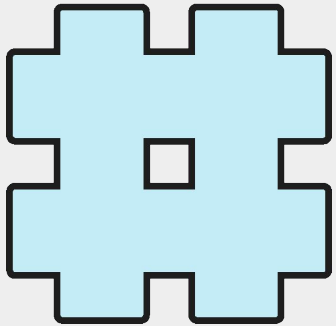


Figure 1: Overview of our approach. We combine a pre-trained retriever (*Query Encoder* + *Document Index*) with a pre-trained seq2seq model (*Generator*) and fine-tune end-to-end. For query x, we use Maximum Inner Product Search (MIPS) to find the top-K documents z_i. For final prediction y, we treat z as a latent variable and marginalize over seq2seq predictions given different documents.

질문 → 검색기(**DPR**) → 문서 → 생성기(**BART**) → 답변

문서를 어떻게 찾을까?

Retriever:
DPR(Dense Passage Retrieval)

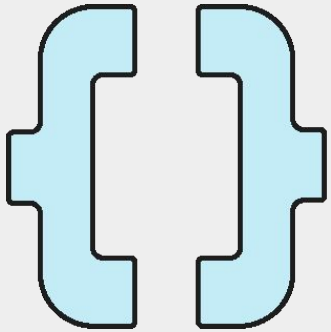


질문(x) → query encoder → q(x)

문서(z) → doc encoder → d(z)

내적 (q(x) · d(z)) → **Top-K 문서 선택 (MIPS)**

질문은 어떻게 만들까?



Generator: BART (Bidirectional and Auto-Regressive Transformers)

입력: 질문 + 선택된 문서(z) (concat, 이어붙이기)

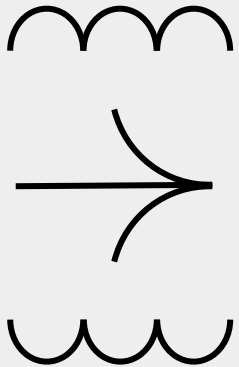
구조: seq2seq encoder-decoder
(입력 문장을 전체 의미를 요약해서 벡터로 저장하고,
그걸 기반으로 새로운 문장을 하나씩 만들어내는 것)

출력: 정답 문장 y 생성

ex) 입력: "I am a student"

인코더: 이 입력을 벡터로 변환

디코더: 벡터를 기반으로 출력 시퀀스를 생성
→ "나는 학생이다."



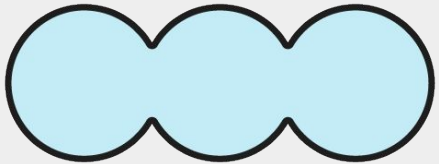
답을 어떻게 고를까?

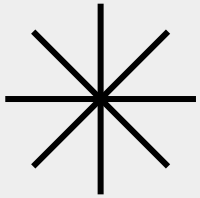
RAG-Sequence

전체 출력 문장을 생성할 때 하나의 문서를 참고

문서별 beam search → 후보 문장 Y

각 문서별 확률 합쳐서 최종 선택





RAG-Token

출력 문장의 단어(토큰)마다 다른 문서를 참고

단어 하나를 만들 때, 여러 문서에서 나온 확률을 평균내어 가장 가능성 높은 단어를 생성하고, 이를 반복해 문장을 만들어낸다.

방식

- Thorough Decoding
  모든 문서에 대해 다시 다 계산하는 방식
- Fast Decoding
  너무 느리니까, 문서에서 한 번도 안 나온 문장은 무시(확률 0) 하는 방식

실험 구성

- 데이터: 위키백과
  (100단어 단위로 쪼개서 2100만 문서)

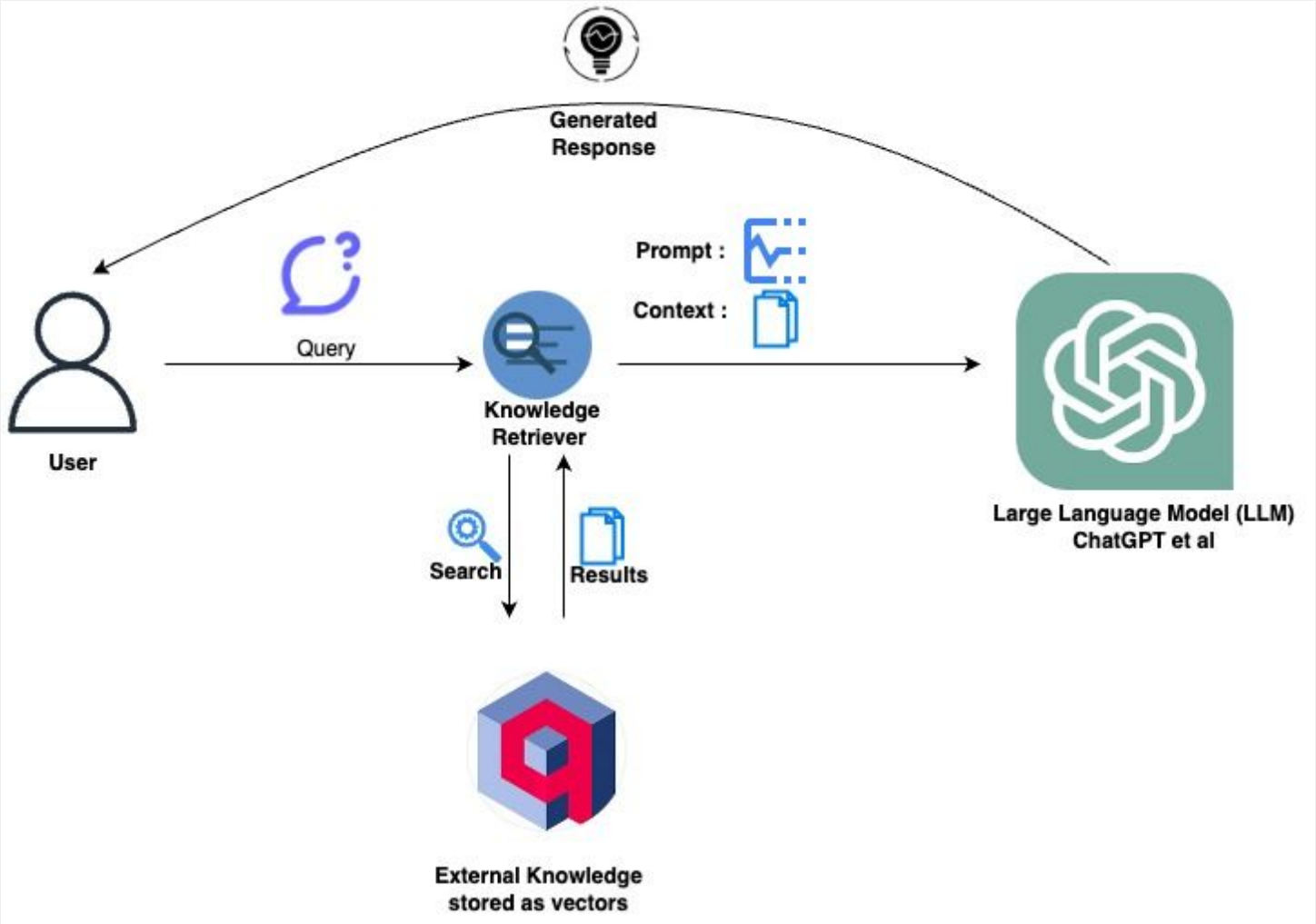
- Top-5, Top-10 문서 검색 실험

- 인덱스

  FAISS
  벡터 기반 MIPS 인덱스 구축

FAISS
Facebook에서 만든 벡터 검색 라이브러리로 어떤 벡터와 가장 비슷한 벡터들을 빠르게 찾는 도구

MIPS
어떤 문서 벡터 $d(z)$와 질문 벡터 $q(x)$ 사이의 내적(inner product)을 계산해서 가장 값이 큰 문서들을 찾는 문제


Generated
Response
Prompt :
Context :
Query
User
Knowledge
Retriever
Large Language Model (LLM)
ChatGPT et al
Search
Results
External Knowledge
stored as vectors

실험 결과

Open-Domain QA

- NQ, TQA, WQ, CT 모두에서 **최고 성능**
- 기존 extractive QA, closed-book QA보다 우수
- 정답 문서에 없어도 **스스로 정답 생성** 가능 (NQ에서 11.8%)

문장 생성 (Abstractive QA / Jeopardy QG)

- RAG는 기존 언어모델(BERT)보다 BLEU, ROUGE, Q-BLEU 모두 높음
- 더 사실적이고 구체적이며, 덜 헛소리함
- 사람이 보기에도 RAG 질문이 사실적이고 구체적

Fact Verification (FEVER)

- RAG는 검색 + 분류도 잘함
- 검색 정확도: Top-1 = 71%, Top-10 = 90%

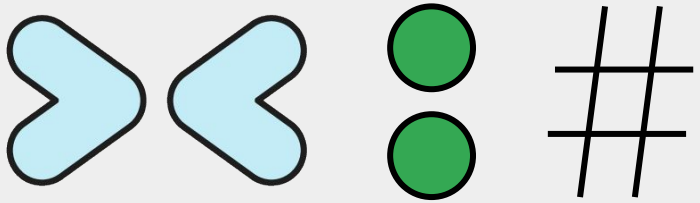
Conclusion

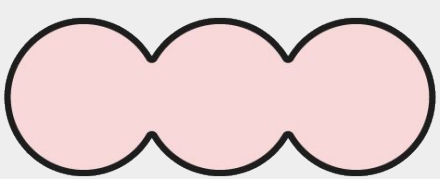
핵심 성과

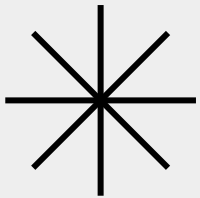
- parametric + non-parametric memory 결합한 “하이브리드 생성 모델”
- RAG: 기존 모델보다 더 사실적이고 구체적인 문장 생성

파라메트릭 메모리
- 모델의 **파라미터**에 지식 저장
- 모델이 훈련 중 얻은 지식은 훈련 이후 변경 X
- 지식의 업데이트 : 모델 재훈련

논파라메트릭 메모리
- **외부 메모리**를 사용하여 실시간으로 정보 검색 -> 결과 생성
- 파라미터 : 고정 / 외부 지식 :동적으로 추가







향후 연구 방향

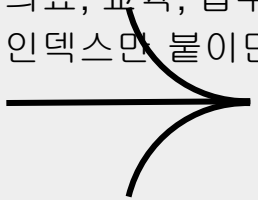
- 다양한 NLP 작업에 확장 적용 기대

Broader Impact

팩트 기반 생성
→ 위키 기반 지식을 활용하여 헛소리(hallucination) 줄임

해석 가능성(Interpretability)
→ 검색한 문서 확인 가능 → 더 투명한 AI

응용 가능성
의료, 교육, 업무지원 등 전문 분야
인덱스만 붙이면 바로 활용 가능

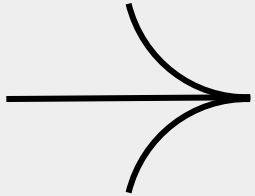


Positive

Broader Impact

편향된 지식 사용 위험
→ 위키 자체도 완벽하지 않음

악용 가능성
- 가짜 뉴스, 피싱, 사칭 등에 악용 가능
- 일자리 대체 가능성 존재



Negative