Retrieval-Augmented Generation

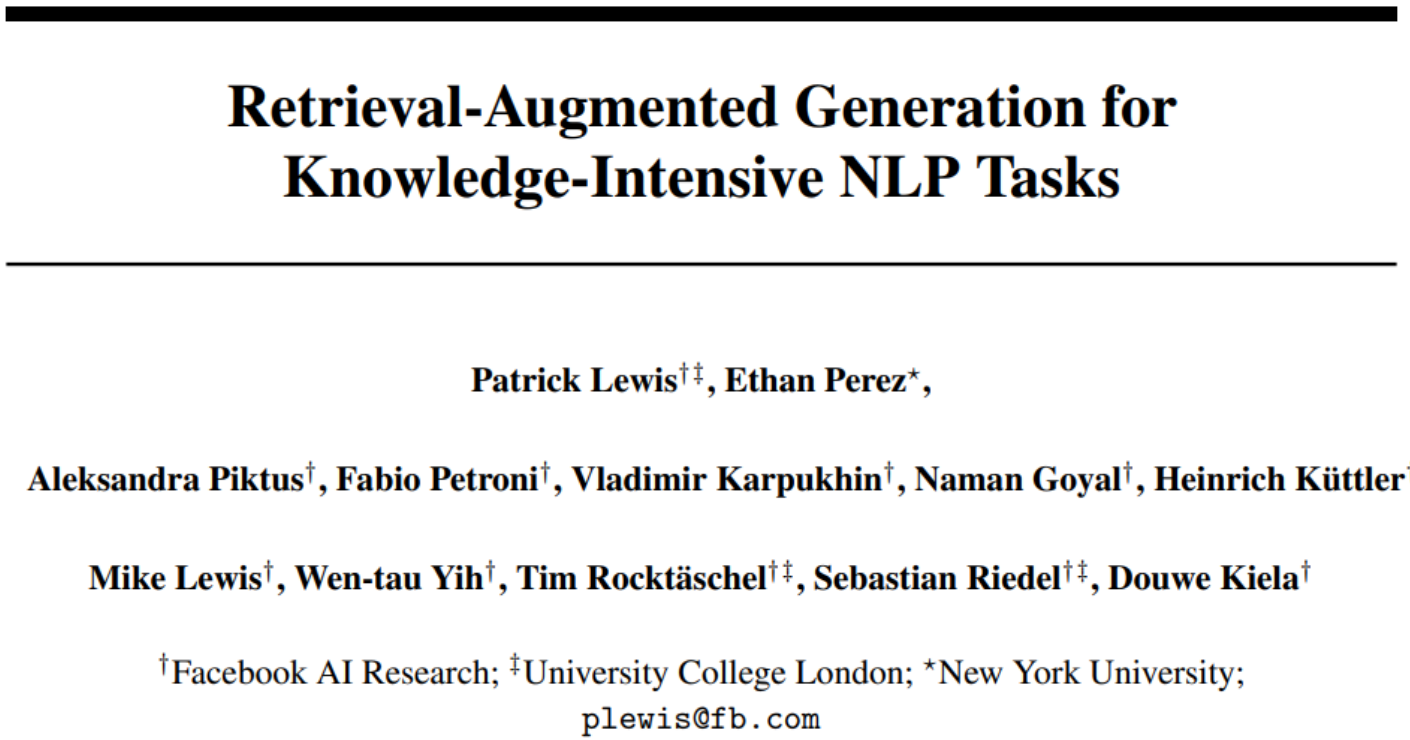
RAG 간단(?) 정리

Index

- RAG (Retrieval Augmented Generation)이 무엇인가
  Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks (NIPS, 2020)
- RAG + LLM
  Retrieval-Augmented Generation for Large Language Model : A Survey (Ongoing, 2023)
- Conclusion

RAG 란 무엇인가

- Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks (NIPS, 2020)



Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks

Patrick Lewis[†‡], Ethan Perez[⋆],

Aleksandra Piktus[†], Fabio Petroni[†], Vladimir Karpukhin[†], Naman Goyal[†], Heinrich Küttler[†],

Mike Lewis[†], Wen-tau Yih[†], Tim Rocktäschel[†‡], Sebastian Riedel[†‡], Douwe Kiela[†]

[†]Facebook AI Research; [‡]University College London; [⋆]New York University;
plewis@fb.com

- RAG 개념이 처음 소개된 논문
- 2024.06.19 기준 인용 2430회

Related Work

- ODQA(Open-Domain QA)
  : Query에 대해, DB에서 관련 문서를 찾아 정답을 알려주는 task

- Knowledge-Intensive Task
  : 사람도 외부 지식 없이 해결하기 어려운 문제

- Parametric, non-parametric Memory (= Implicit, explicit knowledge)
  : 모델에 내재된 지식 / 외부 저장소에 있는 지식

Instruction

Pre-trained Model(PTM) 의 한계

- Pre-trained Model(PTM)은 외부 메모리에 접근하지 않아도 많은 지식 습득
- Implicit knowledge base이기 때문에
  - 메모리 쉽게 확장 및 수정 불가
  - 출처 불분명
  - Hallucination 현상

→ Parametric Memory + non-parametric 을 통해 일부 해결 가능

Instruction

기존 Retrieval base 연구 한계

- Explored Open-domain extractive question answering 에서만 연구

"에베레스트 산의 높이가 어떻게 돼?"

"8,848.86m"

에베레스트산

문서 토론 읽기 편집 역사

에베레스트산(영어: Mount Everest)은 높이가 해발 8,848.86 m로 지구에서 가장 높은 산[1]이다. 네팔에서는 **사가르마타**(산스크리트어: सगरमाथा)[2]라 하고, 티베트어로는 **초모랑마**(티베트어: [tɕʰoɻ.mo˥˥ laŋ˥˥.ma˥˥], 중국조선어: 쵸몰라마봉), 중국어와 문화어에서는 티베트어 '초모랑마'를 그대로 차용해 **주무랑마봉**(중국어: 珠穆朗瑪峰 Zhūmùlǎngmǎ Fēng[*])이라고 부른다. '에베레스트'는 영국의 조지 에버리스트 경의 이름을 따서 붙여졌다. 에베레스트산은 가장 높은 산이지만, 지구의 중심에서 가장 멀리 떨어진 산은 아니다. 지구 중심에서 가장 먼 산은 안데스산맥의 침보라소산이다. 중국과 네팔의 국경이 에베레스트산 정상을 지난다. 에베레스트산괴에는 주위의 로체산(8,516m), 눕체산(7,855m), 창체산(7,580m)이 포함된다.

Instruction

기존 Retrieval base 연구 한계

- Explored Open-domain extractive question answering 에서만 연구

에베레스트산

문서 토론 읽기 편집 역사

에베레스트산(영어: Mount Everest)은 높이가 해발 8,848.86 m로 지구에서 가장 높은 산[1]이다. 네팔에서는 **사가르마타**(산스크리트어: सगरमाथा)[2]라 하고, 티베트어로는 **초모랑마**(티베트어: [tɕʰoɻ.mo˥˥ laŋ˥˥.ma˥˥], 중국조선어: 쵸몰라마봉), 중국어와 문화어에서는 티베트어 '초모랑마'를 그대로 차용해 **주무랑마봉**(중국어: 珠穆朗瑪峰 Zhūmùlǎngmǎ Fēng[*])이라고 부른다. '에베레스트'는 영국의 조지 에버리스트 경의 이름을 따서 붙여졌다. 에베레스트산은 가장 높은 산이지만, 지구의 중심에서 가장 멀리 떨어진 산은 아니다. 지구 중심에서 가장 먼 산은 안데스산맥의 침보라소산이다. 중국과 네팔의 국경이 에베레스트산 정상을 지난다. 에베레스트산괴에는 주위의 로체산(8,516m), 눕체산(7,855m), 창체산(7,580m)이 포함된다.

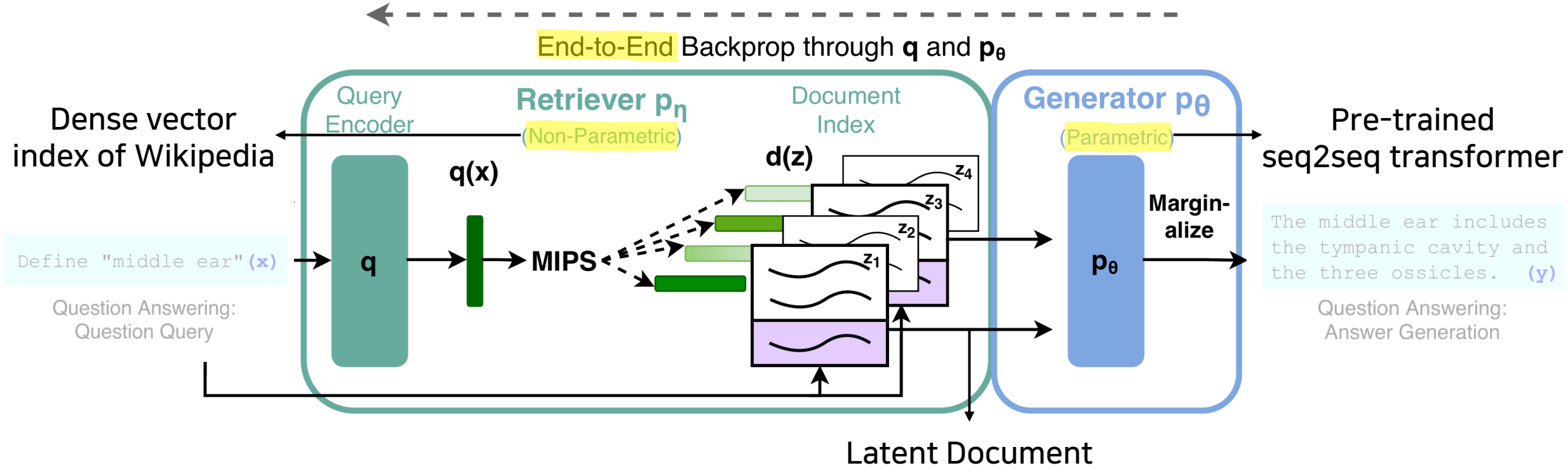
에베레스트산

초모랑마, 사가르마타

Model

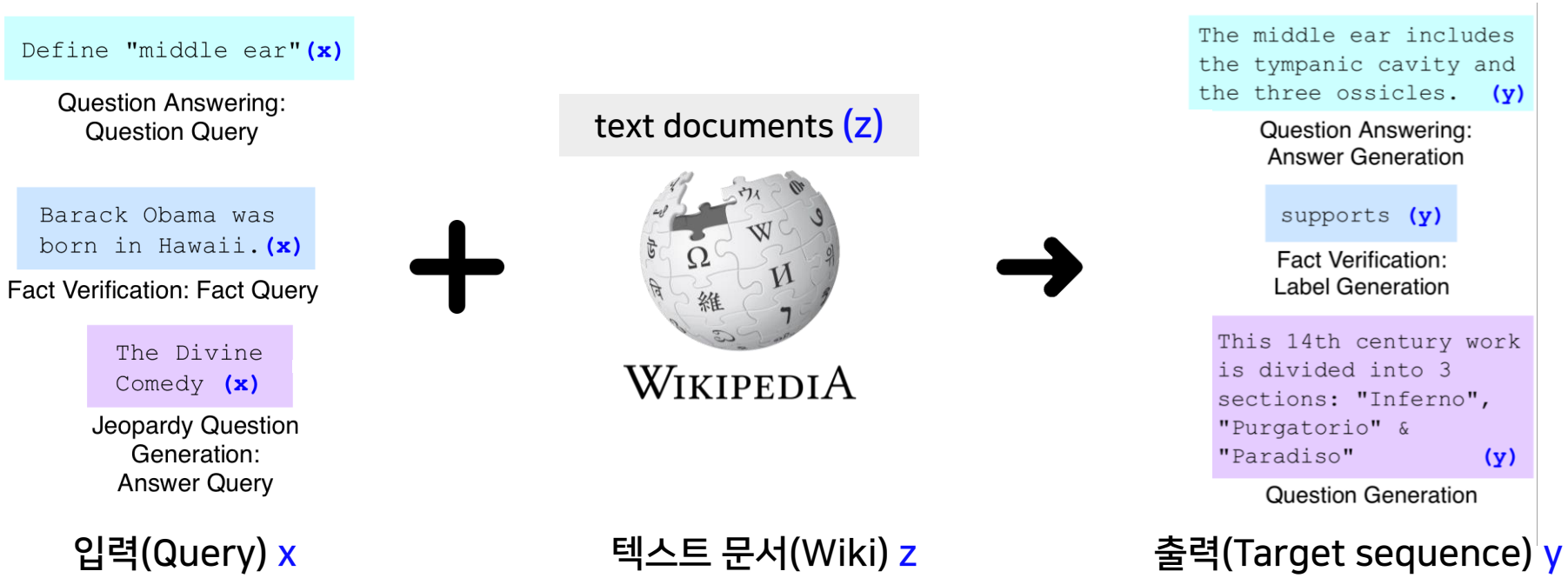
RAG (Retrieval-Augmented Generation)

- Parametric, Non-parametric memory 결합하여 End-to-End로 훈련



Method

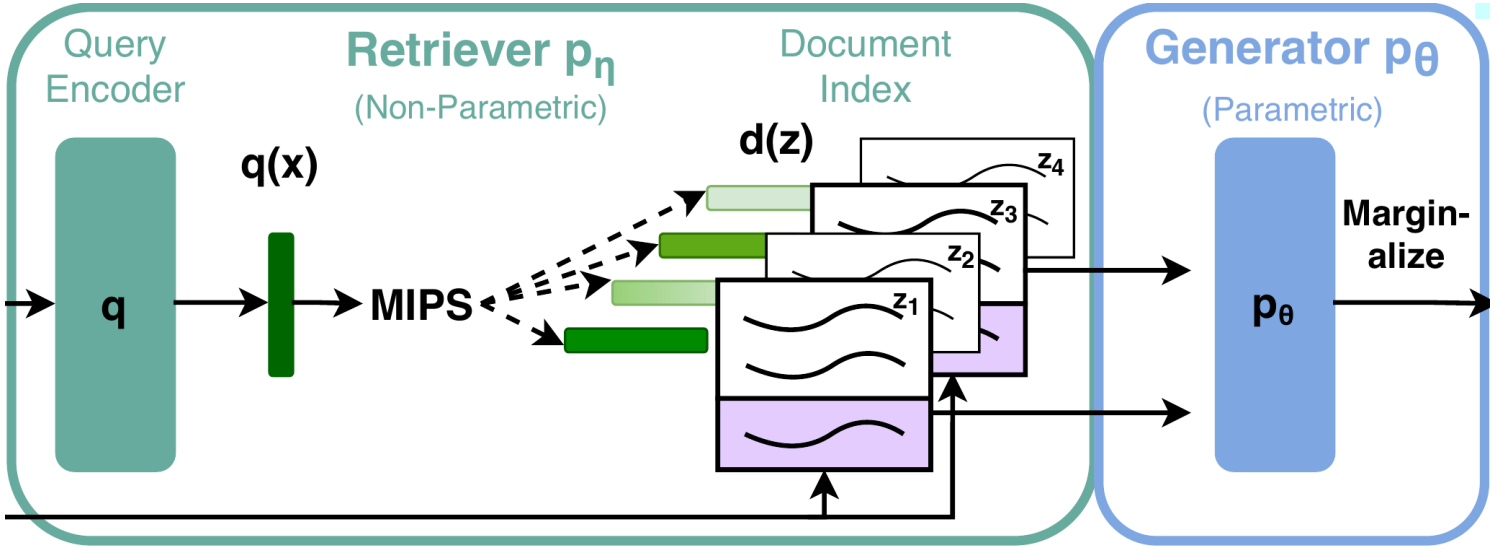
Input / Output



입력 시퀀스 x를 사용하여, z를 검색하고, 검색 결과를 추가적인 context로 사용해, y 생성

Method

Retrieval – Generator 구조

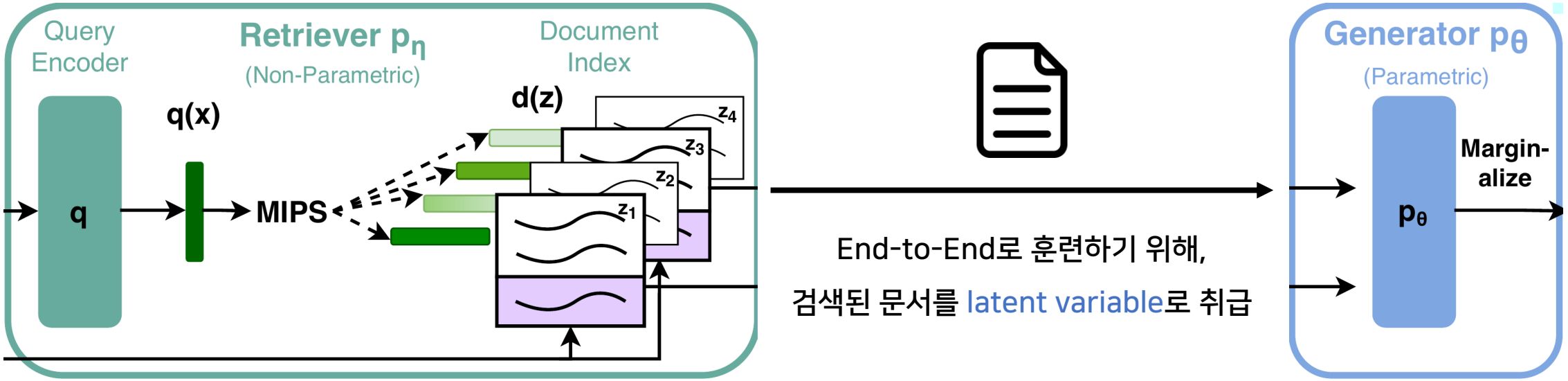


Retrieval : x가 주어졌을 때, text passage에 대한 분포 구한 후, top-k개 반환

Generator : 이전 토큰들과 입력 x, retrieved passage z 기반으로 현재 토큰 생성

Method

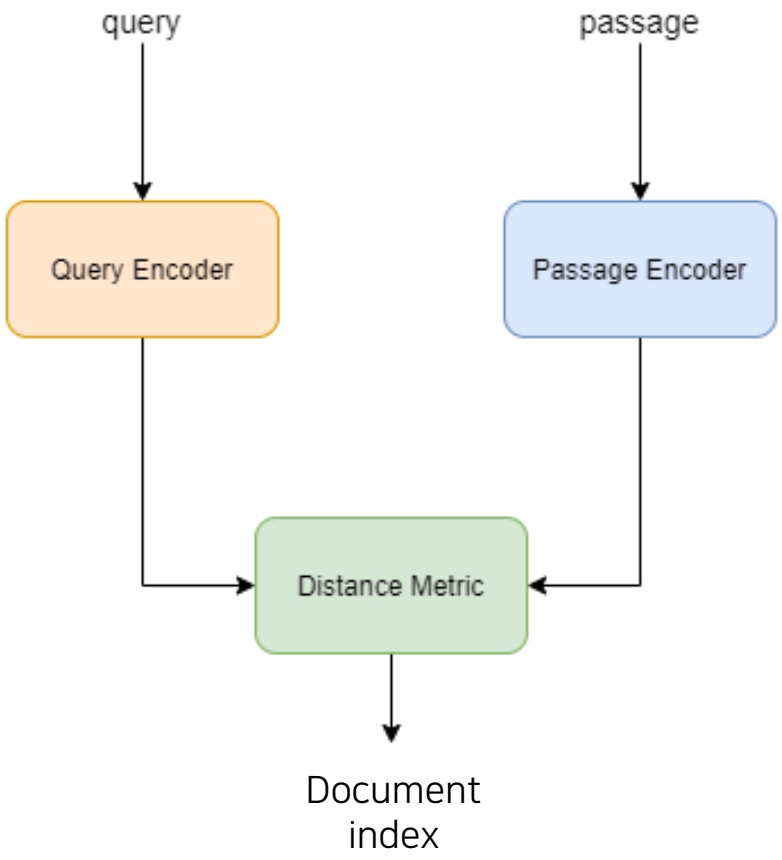
Latent document Marginalization 방식



- RAG-Sequence : 모델이 각 target 토큰 예측하기 위해 동일한 문서 사용
- RAG-Token :각 target 토큰을 다른 문서 기반으로 예측 가능

Retriever: DPR

DPR (Dense Passage Retriever)



Bi-encoder Architecture

- Query encoder
- Document encoder

Generator: BART

- encoder-decoder 모델 어느 것이든 가능 → 본 논문에서는 BART-large 사용.
- BART 써서 생성할 때, input x + 검색한 content z
- Generator 매개변수 = parametric 메모리

Training

- Retriever와 Generator components 훈련 시,

  어떤 문서를 검색 해야하는지에 대한 직접적인 supervision 없이 동시에 훈련

- 입출력 쌍 주어지면 Adam + 확률적 경사 하강법으로 로스 최소화.

- 학습 중, document encoder 업데이트하면 indexing도 같이 업데이트 해야 하기 때문에,

  document encoder 고정하고, query encoder와 generator만 fine-tuning

Decoding

RAG-Token과 Sequence 다른 방식으로 디코딩 진행

- RAG-Sequence Model
  - 하나의 Passage에 대해 끝까지 생성

- RAG-Token Model
  - 하나의 Passage에 대해 토큰 분포 생성

Experiments

- Wikipedia article 100-word로 분할(overlap = 0)하여 21,015,324 document
- DPR document retriever 사용해 임베딩 계산하고, FAISS 통해 single MIPS index 생성
- 학습하는 동안 top-k개 문서 검색. ($k \in \{5,10\}$)

Experiments

데이터셋

- Closed Book에 비해 Open Book 성능 전반적으로 좋음
- RAG 가 REALM, DPR에 비해 성능 좋음

- Baseline

  : 각 데이터셋 별 SOTA + 외부 데이터 활용하지 않은 BERT

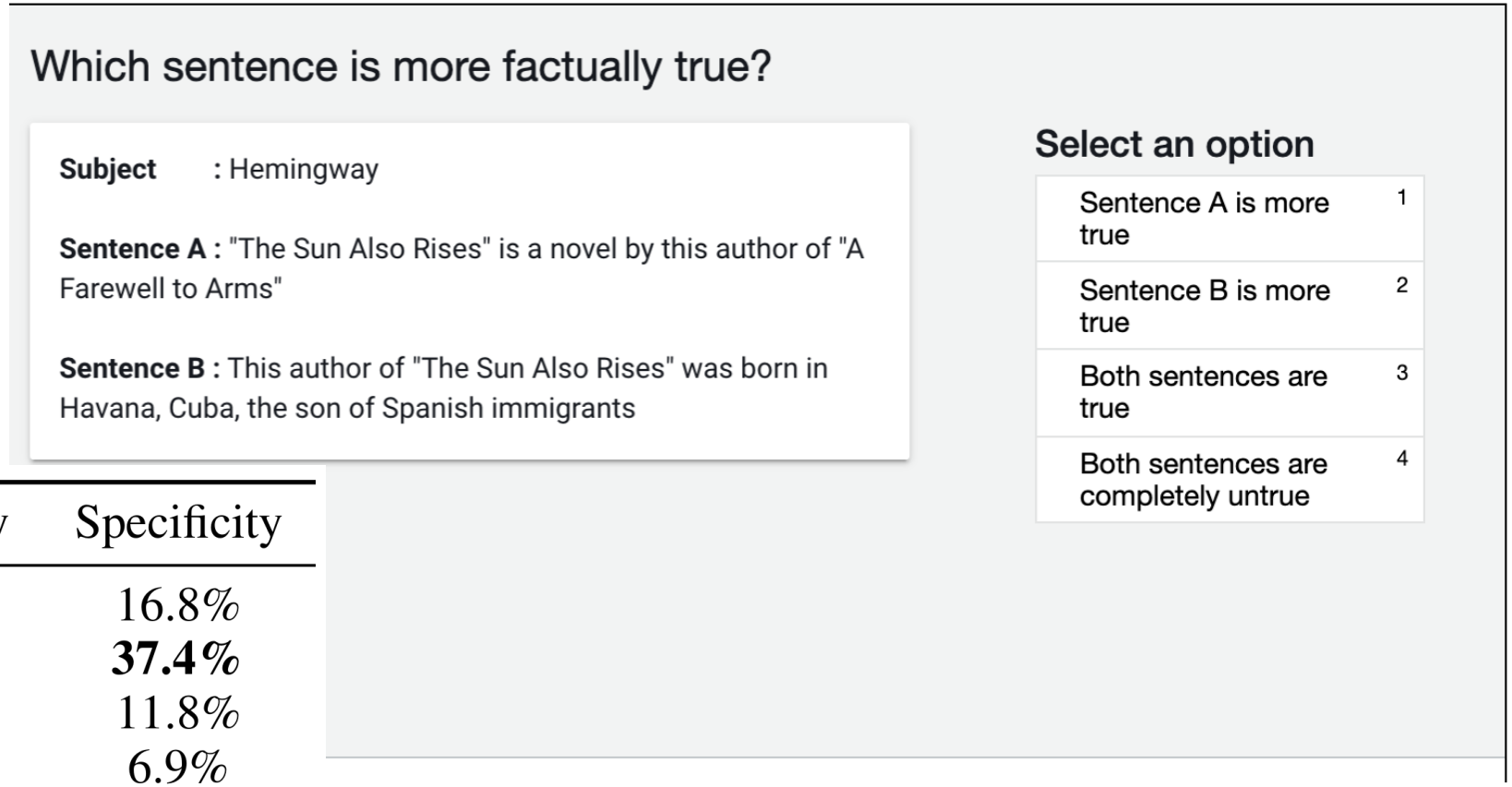
- SOTA에는 못 미치지만, BART보다 좋음

	Model	NQ	TQA	WQ	CT
Closed Book	T5-11B [52]	34.5	- /50.1	37.4	-
	T5-11B+SSM[52]	36.6	- /60.5	44.7	-
Open Book	REALM [20]	40.4	- / -	40.7	46.8
	DPR [26]	41.5	**57.9**/ -	41.1	50.6
	RAG-Token	44.1	55.2/66.1	**45.5**	50.0
	RAG-Seq.	**44.5**	56.8/**68.0**	45.2	**52.2**

Model	Jeopardy B-1	Jeopardy QB-1	MSMARCO R-L	MSMARCO B-1	FVR3 Label Acc.	FVR2 Label Acc.
SotA	-	-	**49.8***	**49.9***	**76.8**	**92.2***
BART	15.1	19.7	38.2	41.6	64.0	81.1
RAG-Tok.	**17.3**	**22.2**	40.1	41.5	72.5	89.5
RAG-Seq.	14.7	21.4	40.8	44.2		

Experiments

Human Evaluation



	Factuality	Specificity
BART better	7.1%	16.8%
RAG better	**42.7%**	**37.4%**
Both good	11.7%	11.8%
Both poor	17.7%	6.9%
No majority	20.8%	20.1%

Contribution

- parametric + non-parametric memory에 접근할 수 있는 하이브리드 생성 모델 제시
- ODQA에서 높은 성능
- BART보다 인간 평가 시, 선호도 높음 (사실적, 구제척)

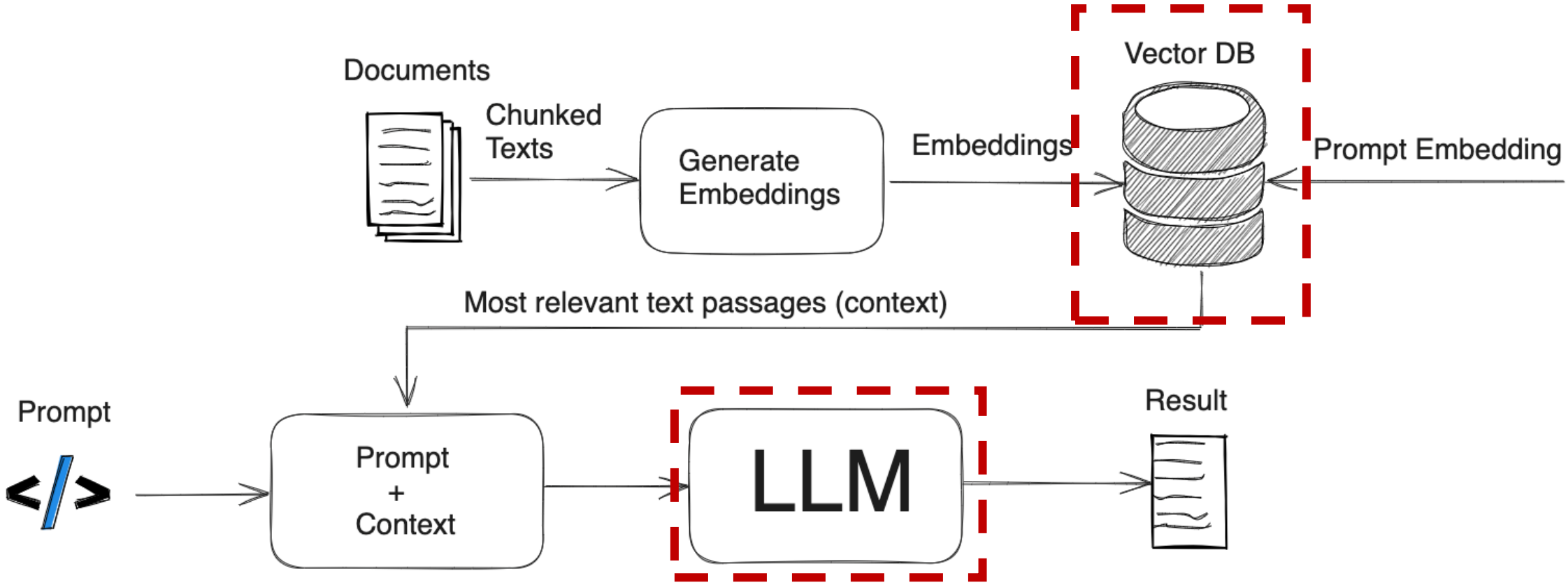
Instruction

Pre-trained Model(PTM) 의 한계

- Implicit knowledge base이기 때문에
  - 메모리 쉽게 확장 및 수정 불가
  - 출처 불분명
  - Hallucination 현상

→ LLM이 겪고 있는 문제

LLMs using RAG



LLMs using RAG : Survey

Retrieval-Augmented Generation for Large Language Models: A Survey

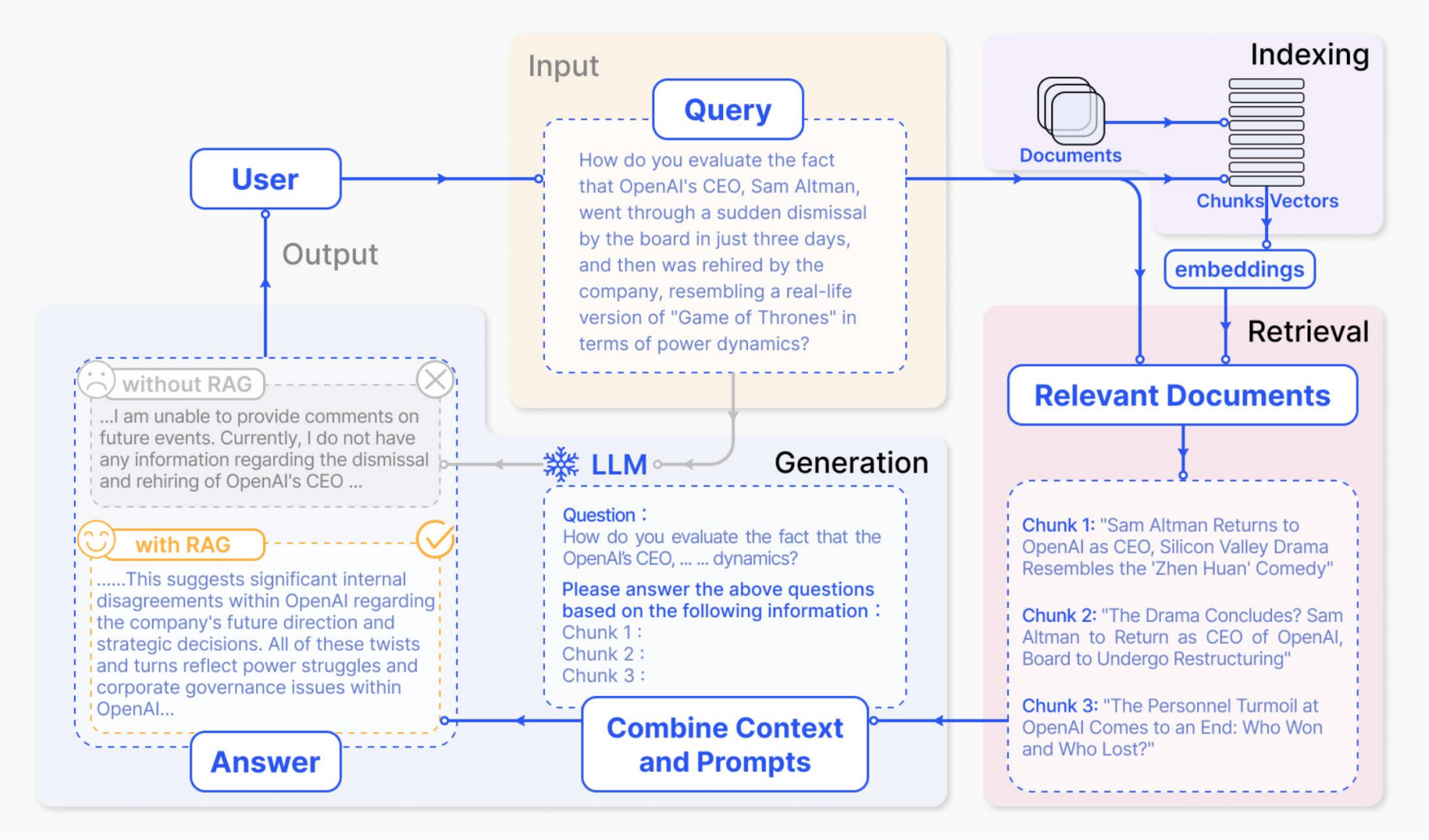
Yunfan Gao[a], Yun Xiong[b], Xinyu Gao[b], Kangxiang Jia[b], Jinliu Pan[b], Yuxi Bi[c], Yi Dai[a], Jiawei Sun[a], Meng Wang[c], and Haofen Wang [a,c]

[a]Shanghai Research Institute for Intelligent Autonomous Systems, Tongji University
[b]Shanghai Key Laboratory of Data Science, School of Computer Science, Fudan University
[c]College of Design and Innovation, Tongji University

***Abstract*—Large Language Models (LLMs) showcase impressive capabilities but encounter challenges like hallucination, outdated knowledge, and non-transparent, untraceable reasoning processes. Retrieval-Augmented Generation (RAG) has emerged as a promising solution by incorporating knowledge from external databases. This enhances the accuracy and credibility of the generation, particularly for knowledge-intensive tasks, and allows**

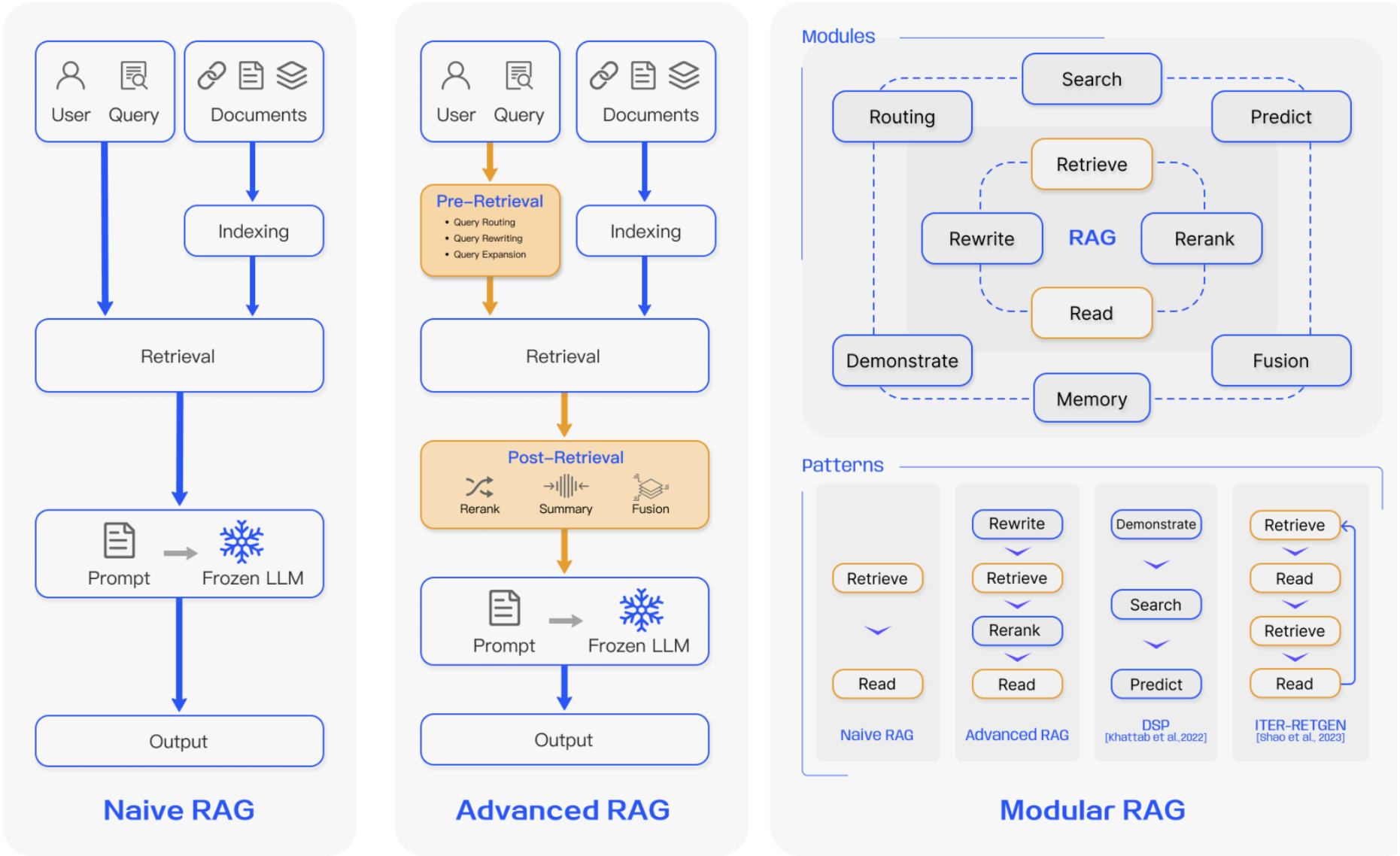
in Figure 1. The development trajectory of RAG in the era of large models exhibits several distinct stage characteristics. Initially, RAG's inception coincided with the rise of the Transformer architecture, focusing on enhancing language models by incorporating additional knowledge through Pre-Training Models (PTM). This early stage was characterized

LLM using RAG Overview

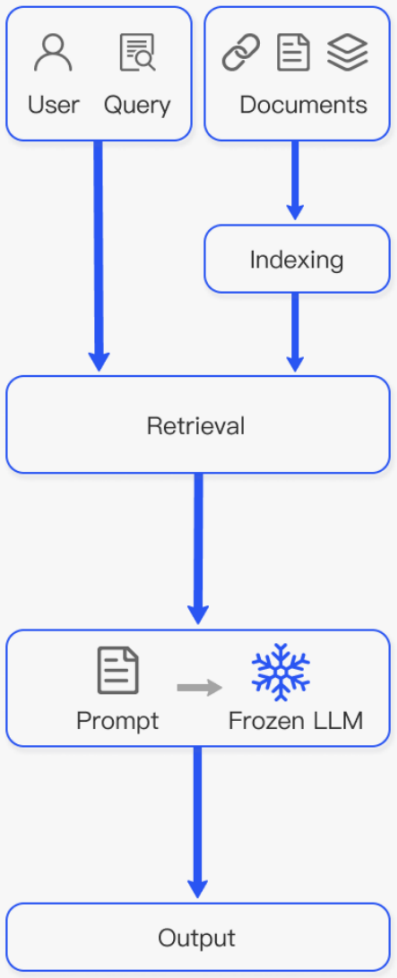


◀ 전형적인 예시

RAG Paradigm



RAG : Naive RAG



Naive RAG

About

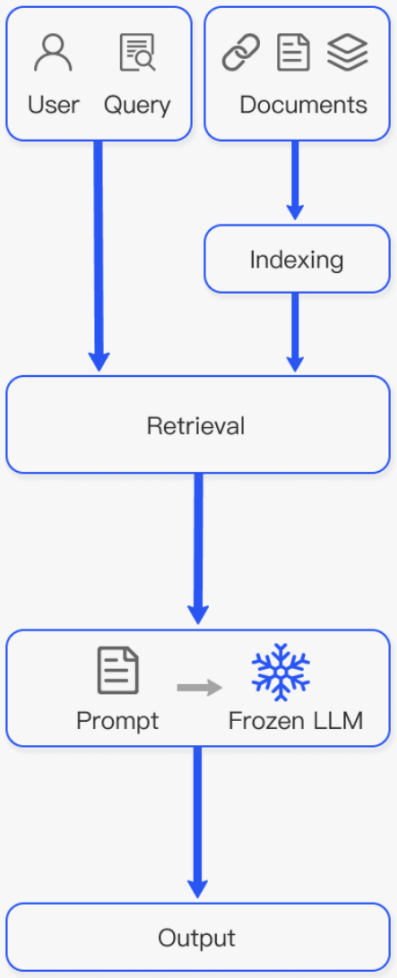
- 인덱싱, 검색, 생성을 포함하는 전통적인 프로세스
- "Retrieve-Read" 프레임워크 라고도 함

Indexing

- PDF, HTML, Word, Markdown과 같은 다양한 형식의 원시 데이터를 정리하고 추출하는 것으로 시작

- LLM에 토큰 제한이 있기 때문에, Chunk로 분할
- Chunk는 임베딩 모델을 통해 벡터 표현으로 인코딩 되어, 벡터 데이터베이스에 저장

RAG : Naive RAG



Naive RAG

Research

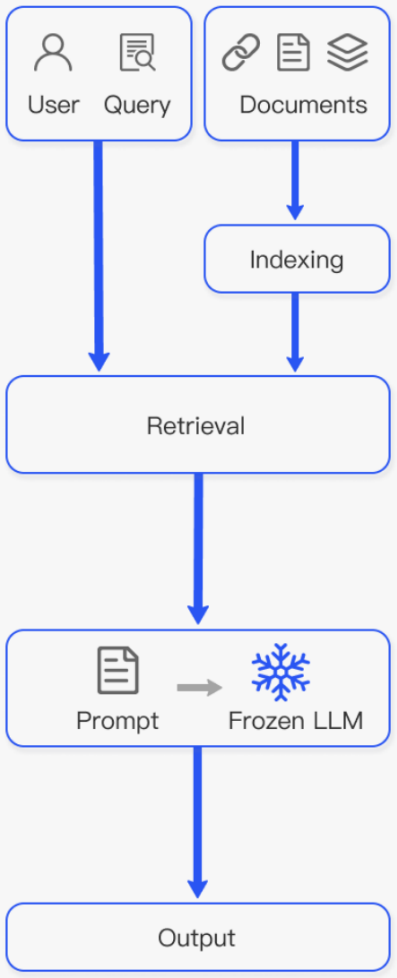
- 쿼리를 벡터 표현으로 변환

- 쿼리 벡터와 인덱싱 된 코퍼스 내 Chunk 벡터 간의 유사성 점수를 계산
- 쿼리와 가장 유사한 top-k개의 Chunk를 우선적으로 검색하여, 프롬프트의 추가 문맥으로 사용

Generation

- Input 쿼리와 top-k 문서를 하나의 프롬프트로 합성하여 답변 생성

RAG : Naive RAG



Naive RAG

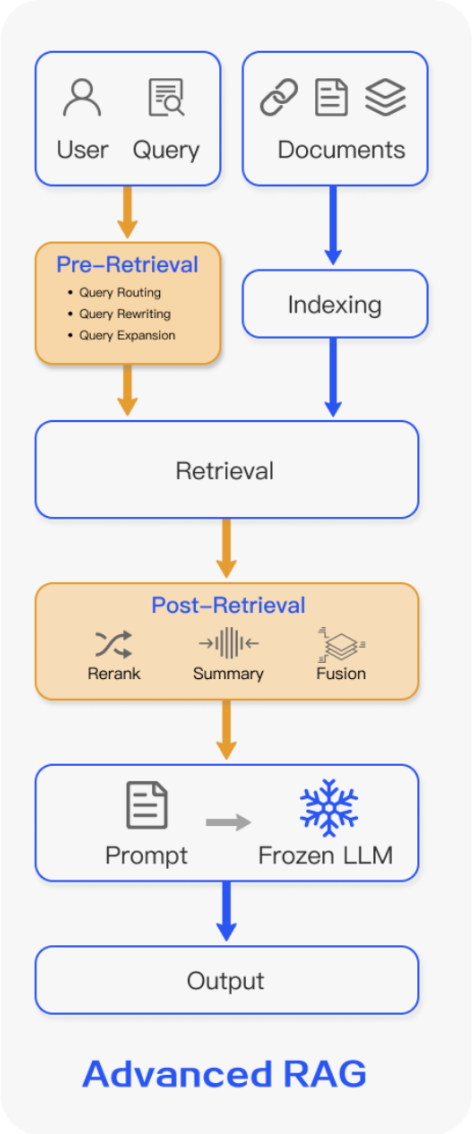
Limitation

1) Retrieval Challenges
   : 잘못 정렬되거나 관련 없는 Chunk 선택 및 중요한 정보 놓치는 경우

2) Generation Difficulties
   : 답변 생성 시, hallucination과 같은 문제

3) Augmentation Hurdles
   : 분절되거나 일관성이 없는 출력 혹은 유사한 정보가 검색될 때 중복성을 겪어 반복적인 응답을 초래

RAG : Advanced RAG



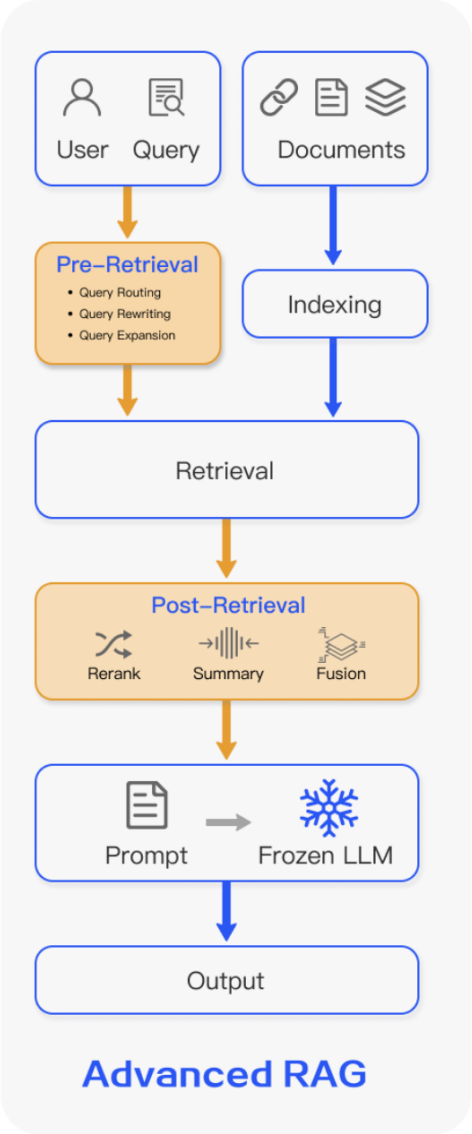
About

- Naive RAG의 한계를 극복하기 위해 사전 검색 및 사후 검색 전략을 활용

Pre-Retrieval Process

- 인덱싱 구조와 원래 쿼리 최적화에 초점
- 인덱싱 최적화 : 데이터 세분화 강화, 인덱스 구조 최적화, 메타데이터 추가, 정렬 최적화, 혼합 검색과 같이 인덱싱 성능 향상
- 쿼리 최적화 : 사용자의 원래 질문을 더 명확하고 검색 작업에 더 적합하게 만드는 것 (쿼리 재작성, 쿼리 변환, 쿼리 확장 등)

RAG : Advanced RAG



About

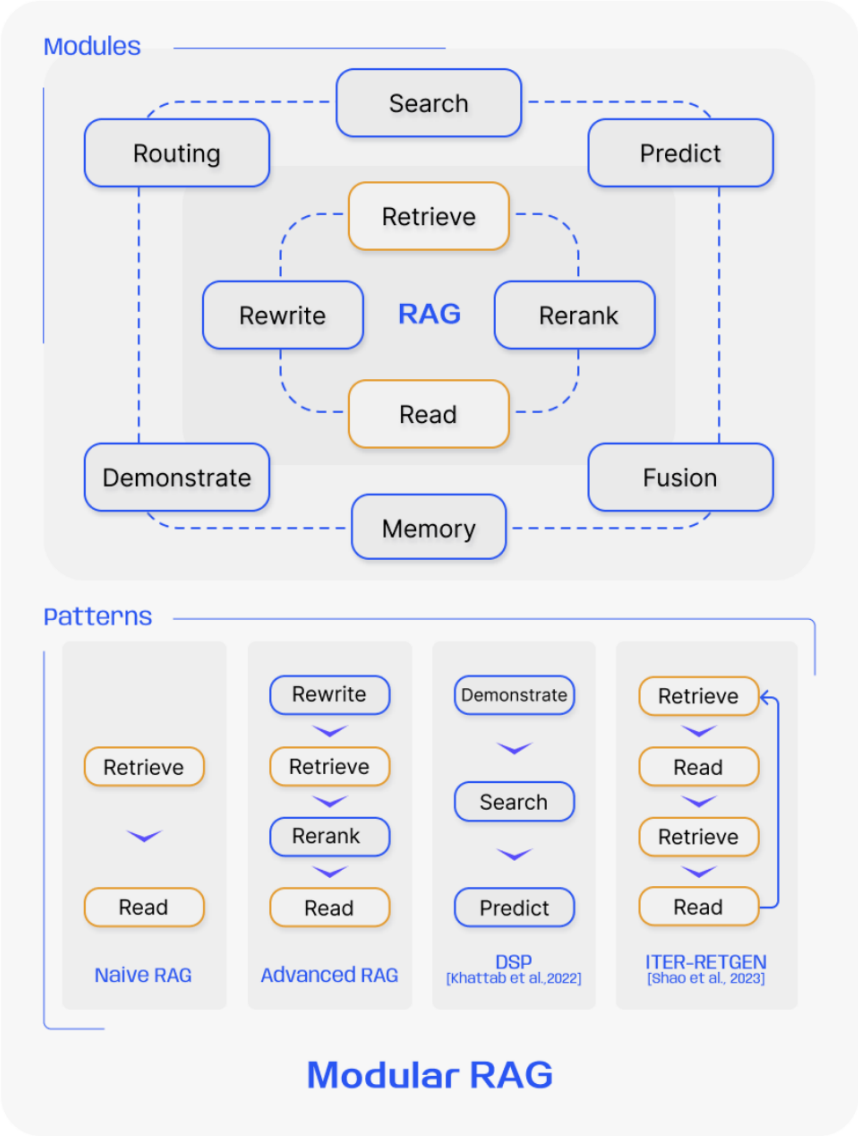
- Naive RAG의 한계를 극복하기 위해 사전 검색 및 사후 검색 전략을 활용

Post-Retrieval Process

- Chunk 재정렬
  : 검색된 정보를 재정렬하여 가장 관련성이 높은 콘텐츠를 프롬프트의 가장 자리에 배치

- 문맥 압축
  : 필수 정보를 선택하고, 중요한 섹션을 강조하며, 처리할 문맥을 축소하는 데 집중

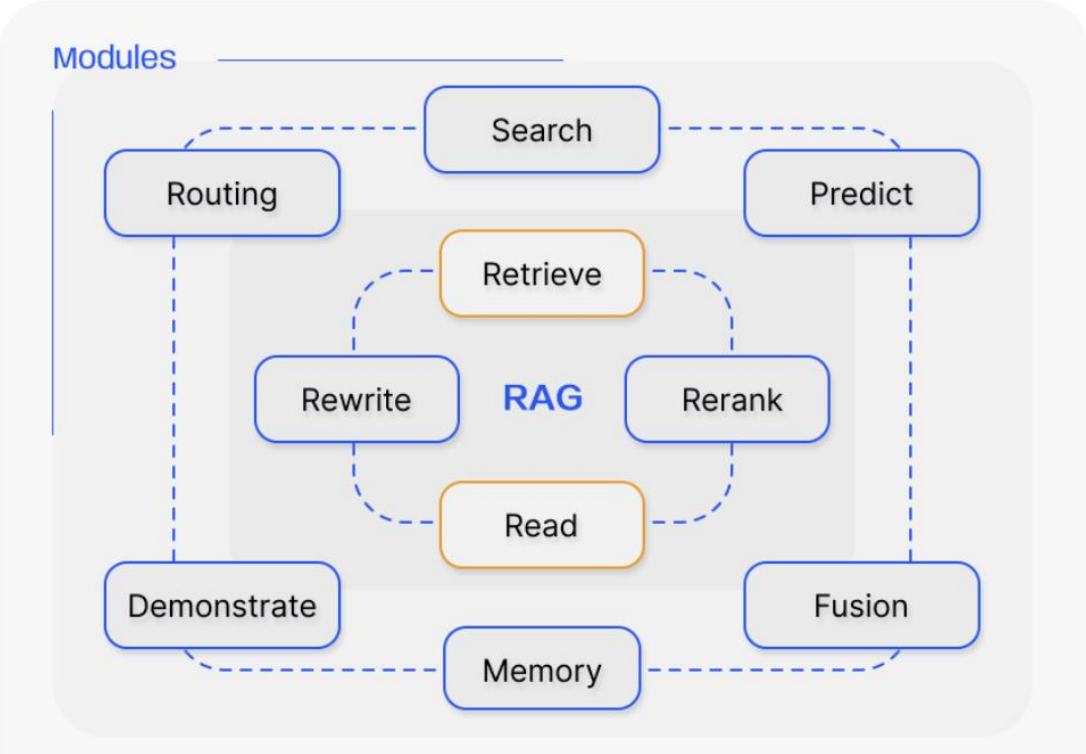
RAG : Modular RAG



About

- Advanced RAG의 발전된 형태
- 다양한 모듈과 기능 통합하여, 더 큰 다양성과 유연성 제공
- 다양한 시나리오와 요구 사항에 맞게 조정할 수 있도록 하는 여러 새로운 모듈과 패턴 포함

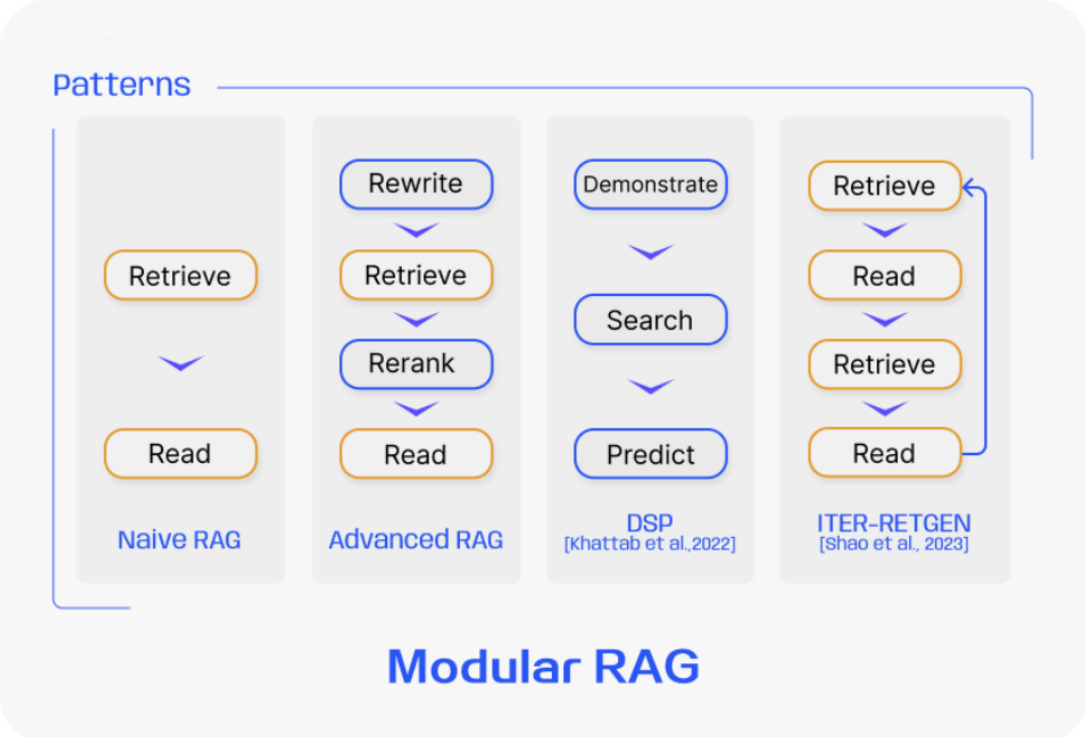
RAG : Modular RAG



New Modules

- 검색 모듈
- 메모리 모듈
- 추가 생성 모듈
- 태스크 적응 모듈
- 정렬 모듈
- 검증 모듈

RAG : Modular RAG



New Patterns

- 모듈 추가 또는 교체
- 모듈 간 흐름 조정

RAG vs Fine-tuning

RAG	Fine-tuning
정밀한 정보 검색 작업에 이상적 (e.g. 정보 검색을 위한 맞춤형 교과서를 모델에 제공하는 것)	특정 구조, 스타일 또는 형식을 복제해야 하는 시나리오에 적합 (e.g. 시간이 지나면서 성장하는 학생)
실시간 지식 업데이트와 높은 해석 가능성으로 외부 지식 소스를 효과적으로 활용하여 동적 환경에 뛰어남	정적이며, 업데이트를 위해 재훈련이 필요하지만 모델의 행동과 스타일 커스터마이징 가능
답변 생성 시, 더 높은 지연 시간	• 데이터셋 준비 및 훈련에 상당한 컴퓨팅 자원을 요구 • 환각을 줄일 수 있지만 익숙하지 않은 데이터에 어려움

Conclusion

- 외부 문서를 입력으로 사용하여, PPT 생성과 비슷한 메커니즘
- 최근 활발한 LLM 연구 동향 파악