



RAG 구성과 운영의 모든 것

searchdoc 백준선

발표자 소개



백준선

- Co-Founder & CEO @ Searchdoc
- ex-AWS, NAVER, IBM

SEARCHDOC



Co-Founder & CEO

Searchdoc

May 2025 - Present · 3 mos

South Korea



Generative AI, Data Architect at AWS

Amazon Web Services (AWS) · Full-time

May 2022 - Present · 3 yrs 3 mos

대한민국 서울

- Led engagement of GenAI chatbot services and RAG-based systems, utilizing sophisticated OpenSearch-based Ranking Models for high-reliability search, founded on client domain knowledge and termi ...see more

▽ Generative AI, Large Language Models (LLM) and +2 skills



NAVER Corp

9 yrs 6 mos

● Feature Engineer at NAVER Search

Dec 2018 - Jun 2022 · 3 yrs 7 mos

대한민국 경기도 성남

- Led research and experiments on ClickModel, a user understanding model utilizing Bayesian probabilistic models, simple LSTM, and self-attention models. Implemented MLE tasks to ensure stable mach ...see more

▽ Information Retrieval, Feature Engineering and +1 skill



NAVER DEVVIEW 2021 SPEAKER

User Understanding for Search Enhancement / 검색 개선을 위한 검색 사용자 이해하기

● Software Engineer at Naver Cloud

Jan 2013 - Dec 2018 · 6 yrs

대한민국 경기도 성남

- Led the development of a Malware Web-document Detection System using ML, conducting research and experiments on modeling based on simple tree models. This involved implementing MLE tasks to ...see more

기존 검색

검색어

nvidia의 주식은 언제 떨어질까

동영상 :

- [사사건건] 엔비디아 액면분할 첫날...주가 전망은? / KBS
- [굿모닝 미주미] 엔비디아만 급등하고 다른 주식은 왜? 엔비

검색결과 출력

생성형 AI기반 검색

검색어

nvidia의 주식은 언제 떨어질까 P사

Pro Search

- NVIDIA의 주식 차트와 역사적 데이터 검색
- NVIDIA의 주식 향후 전망 및 분석 자료를 검색
- NVIDIA의 주식에 대한 전문가들의 의견 검색

Sources

- NVIDIA - 25 Year Stock Price History | NVDA - 1
- Analytical NVIDIA Stock Forecast for 2024, 2025 ... fxopen - 2
- NVIDIA (NVDA) Stock Forecast and Price Target... marketbe... - 3

NVIDIA 주가의 향후 움직임을 정확히 예측하기는 어렵지만, 일부 전문가들은 다음과 같은 이유로 하반기에 단기적인 조정 가능성을 제기하고 있습니다:

- 밸류에이션 부담: NVIDIA의 주가수익비율(PER)이 76배로 매우 높은 수준입니다.¹⁰
- AI 데이터센터 투자 둔화 우려: AI 데이터센터 구축이 일단락되면서 GPU 수요가 줄어들 수 있다는 전망이 있습니다.¹⁰
- 단기 과열 조정: AI 열풍으로 주가가 급등한 후 단기 조정 압력이 있을 수 있습니다.¹⁰
- 경쟁 심화: 다른 기업들의 AI 칩 개발로 NVIDIA의 시장 지배력이 약화될 가능성이 있습니다.
- 거시경제 요인: 금리, 인플레이션 등 거시경제 상황 변화에 따른 영향.

다만 많은 전문가들은 NVIDIA가 AI 시장에서 강력한 입지를 유지하고 있어 장기적으로는 여전히 성장 잠재력이 크다고 보고 있습니다.⁸ 따라서 단기 조정이 있더라도 깊지 않을 것이라는 의견도 있습니다.¹⁰

정보검색,
요약 계획 수립

참고 문서

검색결과
요약 답변

단순 결과 나열에서 정보 요약과 결론 도출까지, 검색의 패러다임이 진화하고 있습니다

혁신적인 대화형 검색이 가능한 이유, 생성형 AI

생성형 AI기반 검색

사용자



↔
자연어 처리
인터페이스

생성형 AI

OpenAI
ANTHROP\IC
Gemini

↔
정보 검색

검색 시스템



사용자



↔

정보 검색

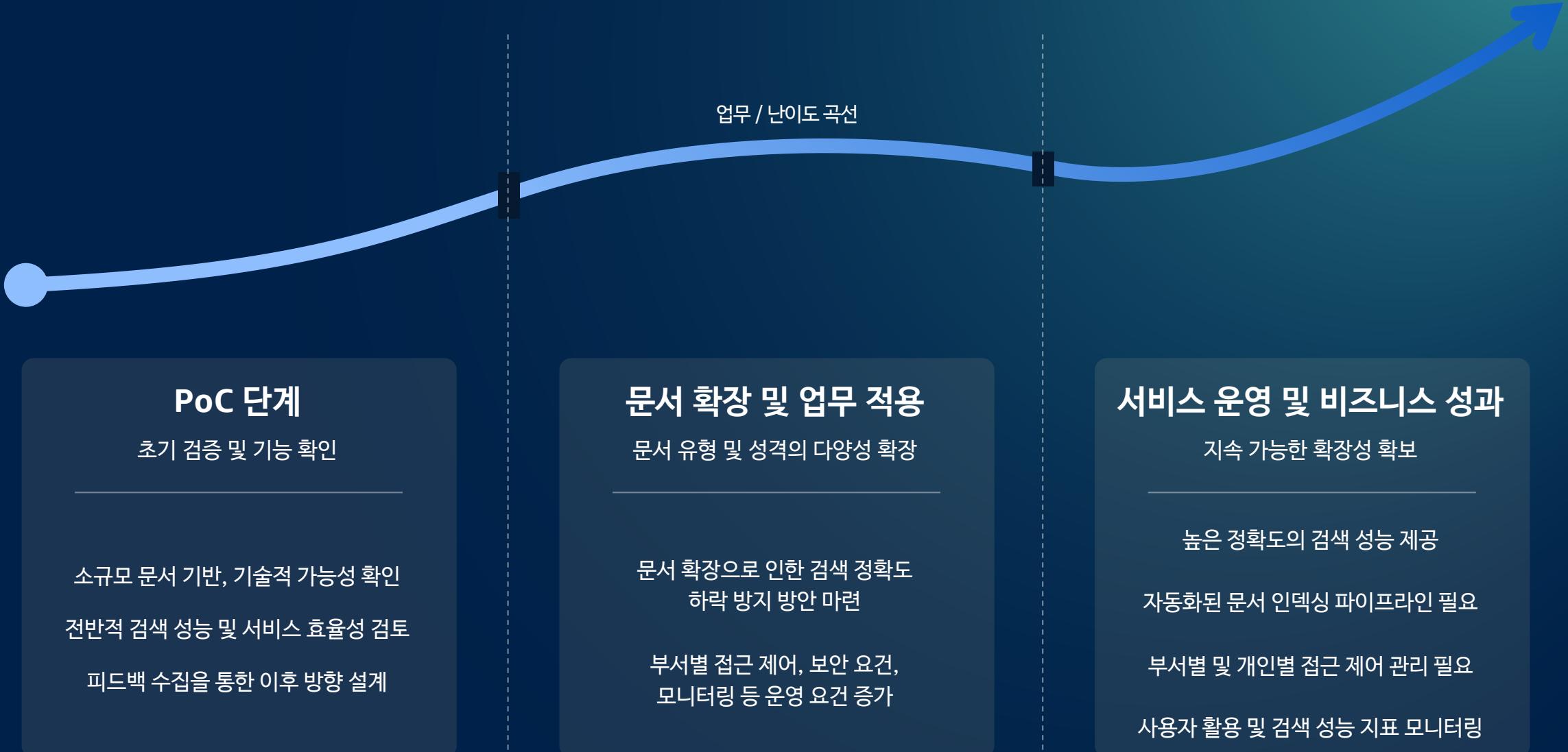
검색 시스템



기존 검색

시장 다변화 속에서도 여전히 어려운 서비스 구축

PoC에서 서비스 운영까지, 높아지는 난이도와 복잡성을 극복하기 위해 체계적인 설계가 필요합니다



Index

- Part 1: 최고의 답변을 찾기 위한 여정
 - Hybrid Search
 - LTR (Learning to Rank)
 - RAG
- Part 2: 똑똑한 RAG를 '제대로' 운영하기
 - Index 전략
 - 접근 제어
 - Observability

Part 0: RAG? Hybrid RAG?

L100-200을 위한 RAG 접근 소개

Why RAG? (1/2)

Without context, answer can easily become a hallucination.

Question

: Can you generate select clause with every columns from the table 'customer_tb'?

Answer

: SELECT customer_id, name, email FROM customer_tb;

Why RAG? (2/2)

With context, answer will be based on context.

Context

```
: CREATE TABLE customer_tb (
    customer_id bigint primary key,
    customer_name varchar(100),
    customer_phone varchar(20),
    register_ymdt timestamp
);
```

Question

: Can you generate select clause with every columns from the table 'customer_tb'?

Answer

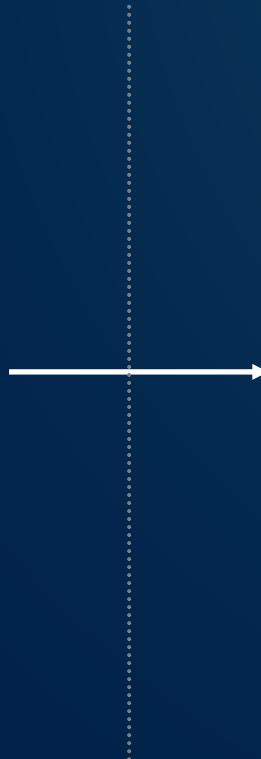
: `SELECT customer_id, customer_name, customer_phone, register_ymdt FROM customer_tb;`

RAG Based on Hybrid Search (1/3)

Non-RAG 구조로, 전달한 Query를 LLM을 활용해 바로 대답하는 구조

Query

출산예정인데,
관련한 휴가를
알려줘



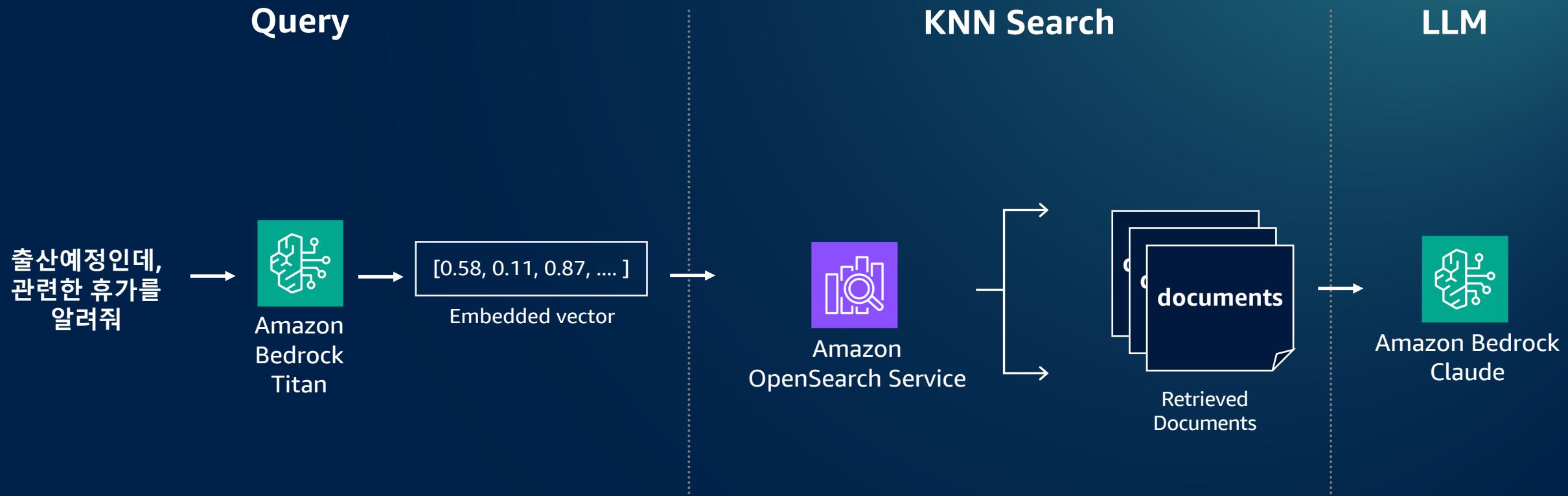
LLM



Amazon Bedrock
Claude

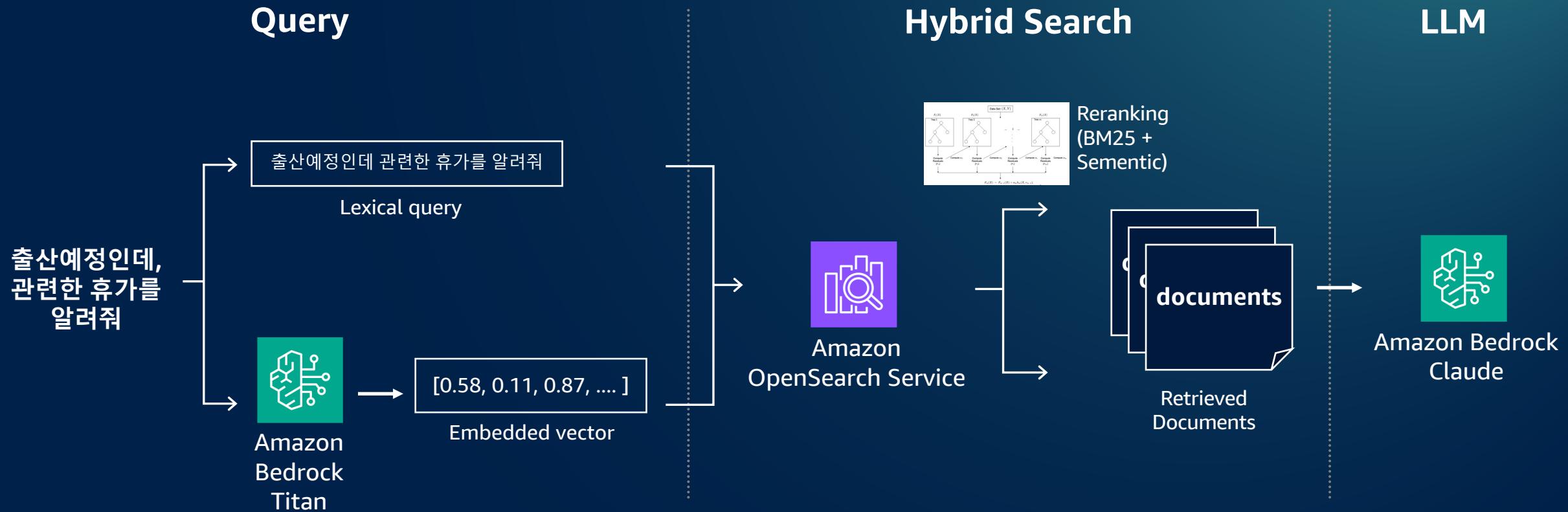
RAG Based on Hybrid Search (2/3)

Embedding Vector를 활용한 Semantic 검색 기반 RAG 구조



RAG Based on Hybrid Search (3/3)

Hybrid Search를 활용한 RAG 구조



Part 1: 최고의 답변을 찾기 위한 예정

Hybrid Search

Better way to retrieve relevant document?

“코끼리 식당”의 검색어는 어느 결과가 더 적합할까요?



Search based on Lexical



Search based on Semantic

Better way to retrieve relevant document?

구분	특징	고려사항
Lexical Search	<ul style="list-style-type: none">- 단어 기반 검색 최적화, 오랜 기간 검증- 동의어, 명사사전 등을 통해 도메인 표현 적용 가능- 초기 구현과 적용이 간편, 성능이 빠르며, 운영 이슈해결이 빠른편	<ul style="list-style-type: none">- 문맥/의미 기반 자연어 이해 부족- 고도화 과정의 러닝커브가 심한편
Semantic Search	<ul style="list-style-type: none">- 대규모 문서로 사전학습된 언어모델 기반- 자연어 기반 문맥/의미를 고려한 검색	<ul style="list-style-type: none">- 사용자 환경의 도메인 이해를 위해, 추가 모델학습 및 운영 필요- 실시간 검색 속도가 비교적 느리며, 운영 이슈해결을 위해선 모델 재학습 필요

Use them all together!

Lexical 검색 및 Semantic 검색을 적절하게 함께 활용하면, 여러 장점을 기반하여 서비스 가능한 검색 환경을 구축하기 용이

Lexical
Search + Semantic
Search =

- 단어 및 문맥 기반의 종합적인 검색
- 도메인 활용 단어 및 표현을 기반하여, 검색 운영 가능
- 운영간 랭킹관련 이슈 해결이 비교적 수월

Use them all together!

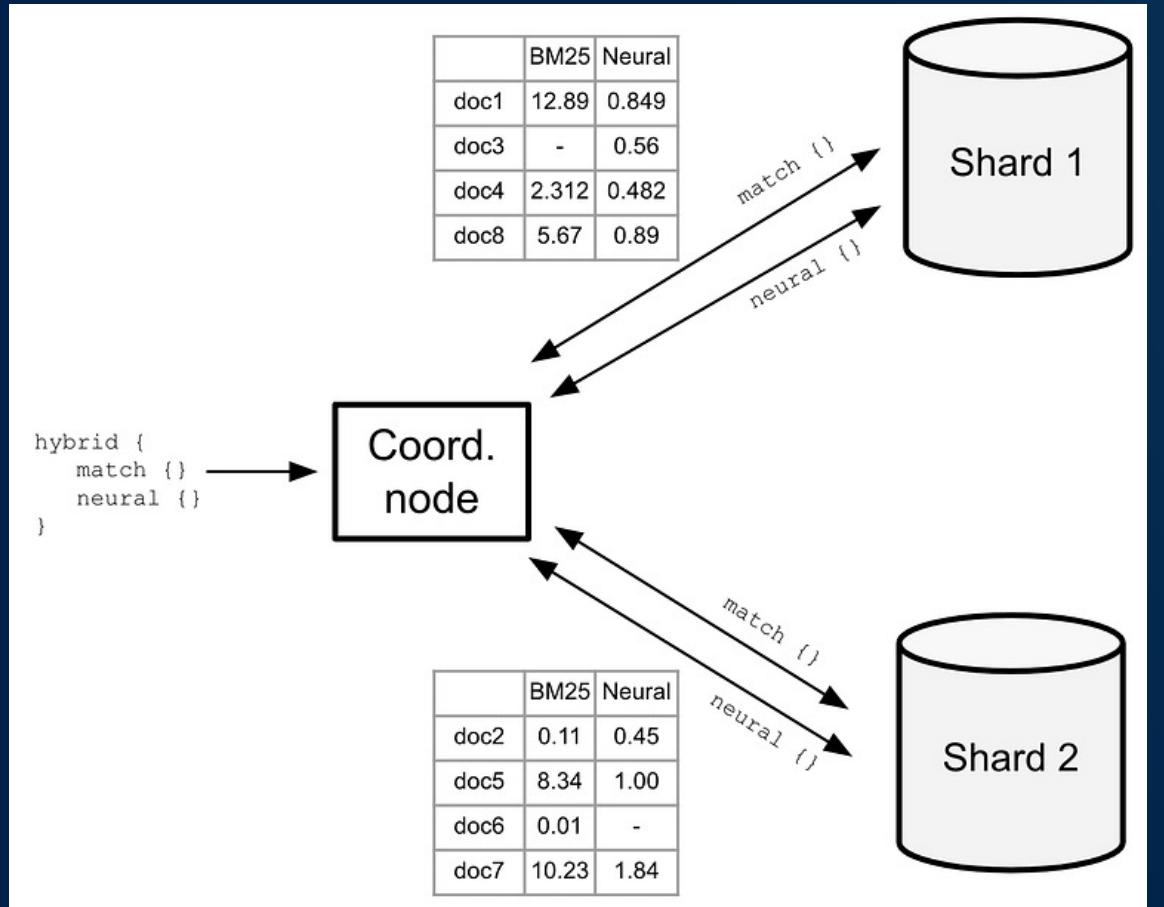
Lexical 검색 및 Semantic 검색을 적절하게 함께 활용하면, 여러 장점을 기반하여 서비스 가능한 검색 환경을 구축하기 용이

Lexical
Search + Semantic
Search =

Hybrid Search

(aka. Ensemble Search, Reranking, ...)

Hybrid Search : How to score? (1/4)



-	Lexical (bm25)	Semantic (knn, neural)	Hybrid
doc1	12.89	0.849	?
doc2	0.11	0.45	?
doc3	-	0.54	?
doc4	2.312	0.482	?
...	?

<https://opster.com/guides/opensearch/opensearch-machine-learning/opensearch-hybrid-search/>



Hybrid Search : How to score? (2/4)

-	Lexical (bm25)	Semantic (knn, neural)
doc1	12.89	0.849
doc2	0.11	0.45
doc3	-	0.54
doc4	2.312	0.482
...



- Type1 :
: **bm25 + knn**
- Type2 :
: **bm25 * weight1 + knn * weight2**
- Type3 :
: **normalize(bm25) + normalize(knn)**
- Type4 :
: **normalize(bm25) * weight1 + normalize(knn) * weight2**
- Type5 :
: **normalize(weight1 * bm25) * weight2 + filter(knn) * weight3**

Hybrid Search : How to score? (3/4)

- Type1: Too naive, 각 필드의 Scale이 다르기 때문에, 수식 자체의 이슈 존재.
: ~~bm25 + knn~~
- Type2: Weight이 적절하다면 좋은 Ranking 기대 가능. 최적의 Weight 실험 필요.
: $bm25 * weight1 + knn * weight2$
- Type3: Scale 맞춰짐. 어느정도의 Ranking 기대 가능하지만, 각 필드의 중요도를 가할 수 없음.
: ~~normalize(bm25) + normalize(knn)~~
- Type4: Scale 맞춰짐, 필드의 중요도 포함. 필요한 수준의 Ranking 기대 가능.
: $normalize(bm25) * weight1 + normalize(knn) * weight2$
- Type5: 필요한 모든 로직 반영 가능. 최적의 수식 실험 필요.
: $normalize(weight1 * bm25) * weight2 + filter(knn) * weight3$

Hybrid Search : How to score? (4/4)

- Type1 :

: ~~bm25 + knn~~

- Type2 :

: $\text{bm25} * \text{weight1} + \text{knn} * \text{weight2}$

- Type3 :

: ~~normalize(bm25) + normalize(knn)~~

- Type4 :

: $\text{normalize}(\text{bm25}) * \text{weight1} + \text{normalize}(\text{knn}) * \text{weight2}$

- Type5 :

 OpenSearch : $\text{normalize}(\text{weight1} * \text{bm25}) * \text{weight2} + \text{filter}(\text{knn}) * \text{weight2}$

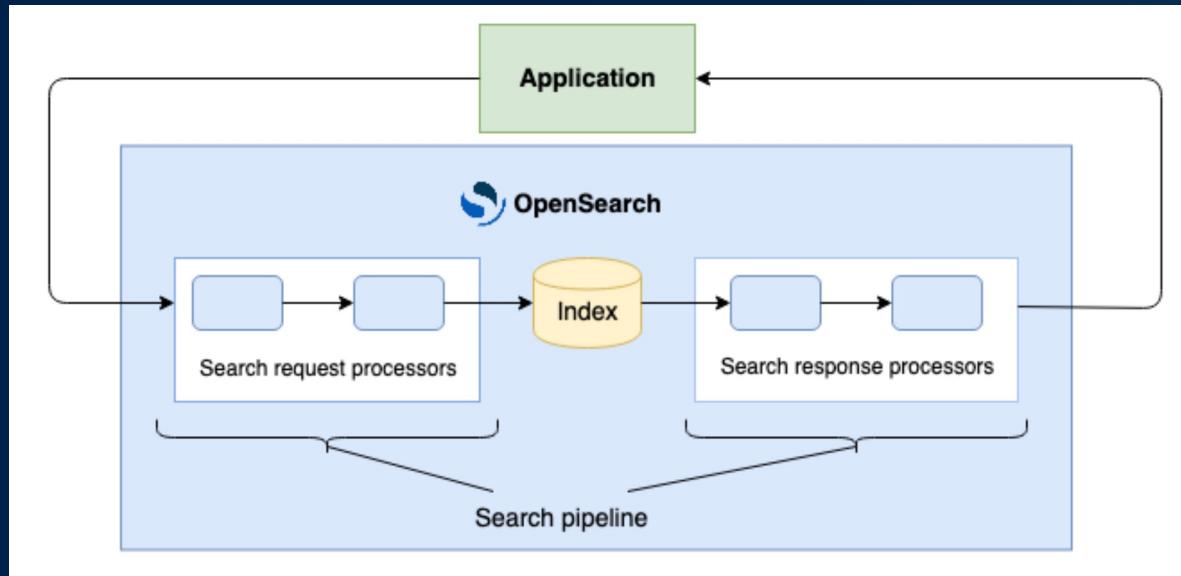
》 Learning To Rank 혹은
Function Score를 통한 구현

》 OpenSearch : Search Pipeline을 통한 구현

》 Learning To Rank 혹은
Function Score를 통한 구현

OpenSearch : Search Pipeline

Efficient Solution for Processing Search Queries and Results



1. Search Request Processor

- Query 및 Request에 포함된 메타데이터 활용
- 검색 전 작업을 수행하고 검색 요청을 반환

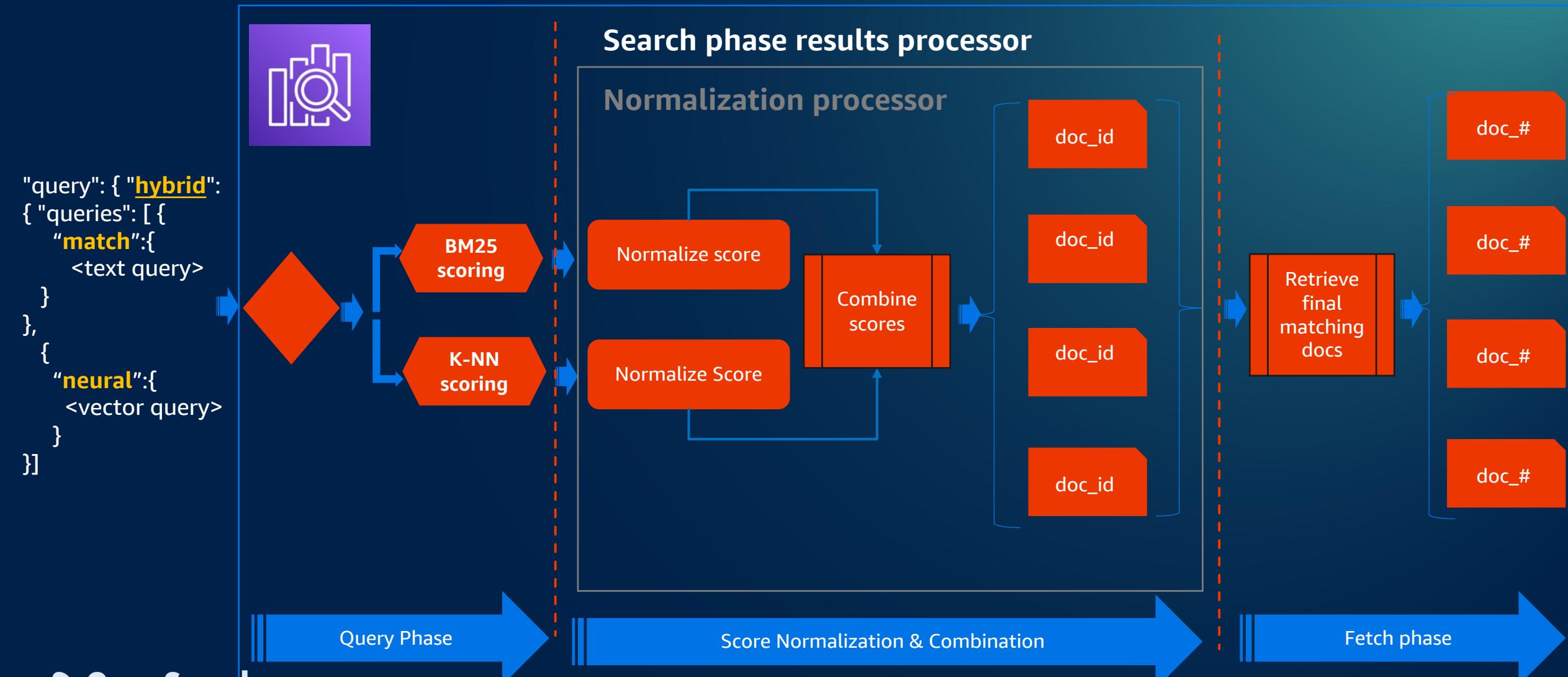
2. Search Response Processor

- 검색 결과 및 Request의 메타데이터를 활용
- 검색 대상 문서를 기준, 최종 Ranking 요소에 대한 추가 작업을 수행하여 문서를 반환

3. Search phase results processors

- 검색 결과 문서를 기반, 최종 결과 생성 전 여러 값의 변형을 가하는 방안 활용
- 검색 결과 및 Request의 메타데이터를 활용하여 검색 대상 문서를 처리

OpenSearch : Search Pipeline > Result Pipeline



Building Hybrid Search

Building hybrid search pipeline

```
PUT /_search/pipeline/nlp-search-pipeline-hybrid
{
  "description": "Post processor for hybrid search",
  "phase_results_processors": [
    {
      "normalization_processor": {
        "normalization": {
          "technique": "min_max"
        },
        "combination": {
          "technique": "arithmetic_mean",
          "parameters": { "weights": [ 0.4, 0.6 ] } }
      }
    }
  ]
}
```

Request with Search Pipeline

```
GET /sm_semantic_demostore/_search?search_pipeline=nlp-search-pipeline-hybrid
{
  "_source": {
    "includes": [
      "name", "description"
    ]
  },
  "query": {
    "hybrid": {
      "queries": [
        {
          "match": {
            "text": {
              "query": "red shirt"
            }
          }
        },
        {
          "neural": {
            "name_v": {
              "query_text": "red shirt",
              "model_id": "OSmmWYwB33bA6yQYMuag",
              "k": 5
            } } } } } }
```

field - 1

field - 2

Search Pipeline Result Sample

A	B	C	D	E	F	G	H	I
		BM25	BM25 normalized	BM25 weighted weight: 0.4	Neural	Neural normalized	Neural weight: 0.6	Combined score rank
doc1	12.890	1.000	0.400	1.63	0.849	0.509	0.455	2nd
doc2	0.110	0.008	0.003	0.45	0.000	0.000	0.002	
doc3		0.000	0.000	0.56	0.079	0.047	0.024	
doc4	2.312	0.179	0.072	1.12	0.482	0.289	0.180	
doc5	8.340	0.647	0.259	1.00	0.396	0.237	0.248	3rd
doc6	0.010	0.000	0.000	0.000	0.000	0.000	0.000	
doc7	10.230	0.793	0.317	1.84	1.000	0.600	0.459	1st
doc8	5.670	0.439	0.176	0.89	0.317	0.190	0.183	

<https://opster.com/guides/opensearch/opensearch-machine-learning/opensearch-hybrid-search/>

Finally!!

“코끼리가 들어가고 있는 코끼리 식당”을 검색해낼 수 있습니다.



Search based on Hybrid Search

Part 1: 최고의 답변을 찾기 위한 예정

Learning To Rank

Think about the search results again

"코끼리가 들어가고 있는 코끼리 식당"이 진정 우리가 원하는 식당일까요?



별점 1.4 (리뷰 7건)



별점 4.8 (리뷰 672건)

Determining Relevant Document



Similarity

단어, 키워드 기반
문맥 의미 기반



Quality

문서의 작성 퀄리티
문서 작성자의 신뢰도



User behavior

사용자 조회 수
좋아요/싫어요 비율
상주기간, 이탈율

How to score with multiple features? (1/3)

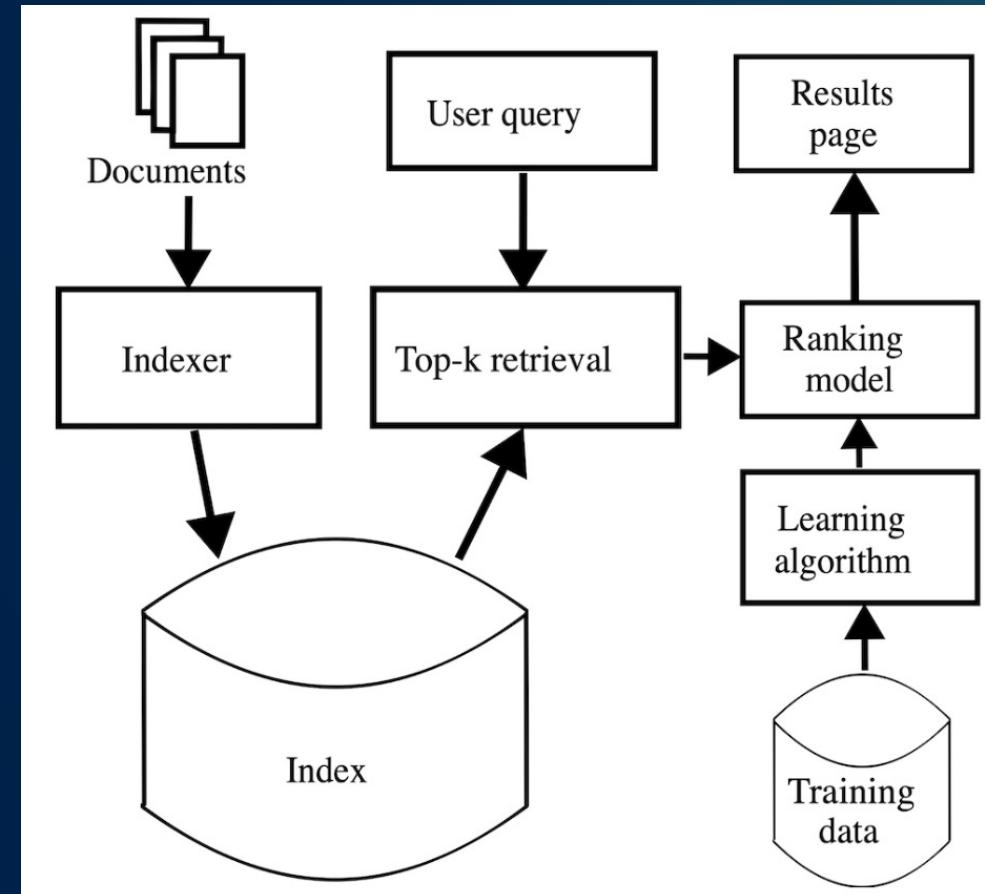
-	Similarity Features			Quality Feature	User Behavior Feature	Combined Score
	Lexical - 1 (bm25)	Lexical - 2 (bm25)	Semantic - 1 (knn, neural)			
doc1	12.89	6.1	0.15	0.12	4.75	?
doc2	0.11	0.31	0.08	1.56	3.51	?
doc3	-	0.41	0.75	9.75	3.61	?
doc4	2.312	0.125	0.77	0.81	-	?
...	?

How to score with multiple features? (2/3)

- Type1 :
: **bm25_1** + **bm25_2** + **bm25_3** + **knn_1** + **knn_2**
- Type2 :
: **bm25_1** * **w1** + **bm25_2** * **w2** + **bm25_3** * **w3** + **knn_1** * **w4** + **knn_2** * **w5**
- Type3 :
: **norm(bm25_1)** + **norm(bm25_2)** + **norm(bm25_3)** + **norm(knn_1)** + **norm(knn_2)**
- Type4 :
: **norm(bm25_1)** * **w1** + **norm(bm25_2)** * **w2** + **norm(bm25_3)** * **w3** + **norm(knn_1)** * **w4**
- Type5 :
: **norm(w1 * bm25_1)** * **w2** + **norm(bm25_2)** * **w3** + **bm25_3** * **w4** + **filter(knn_1)** * **w5**

How to score with multiple features? (3/3)

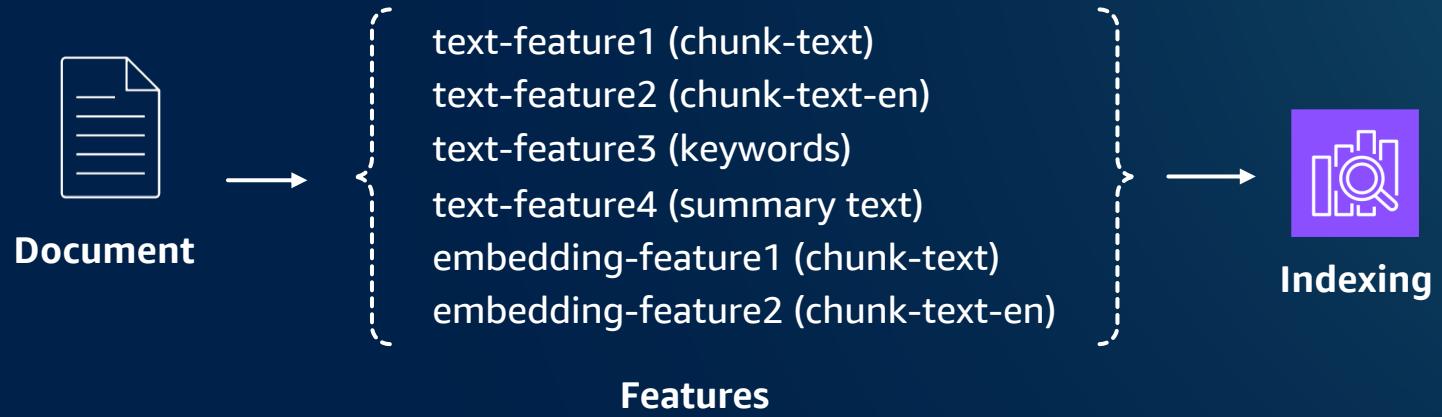
Now, Time to Learning To Rank!



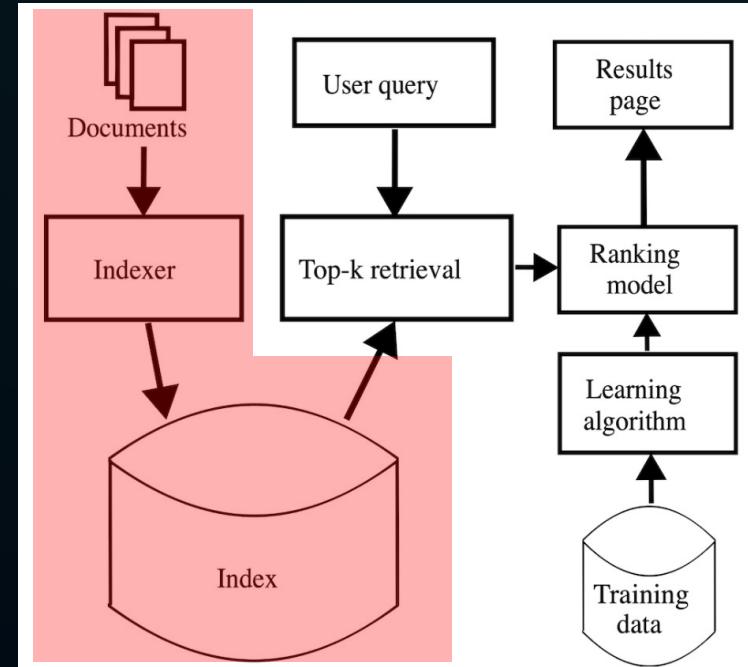
https://en.wikipedia.org/wiki/Learning_to_rank

How to LTR (1/3)

Indexing Document



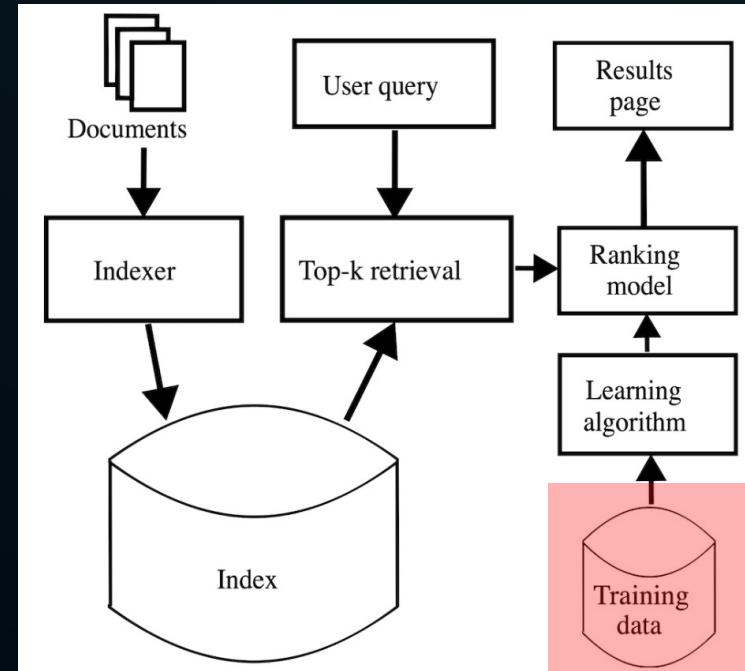
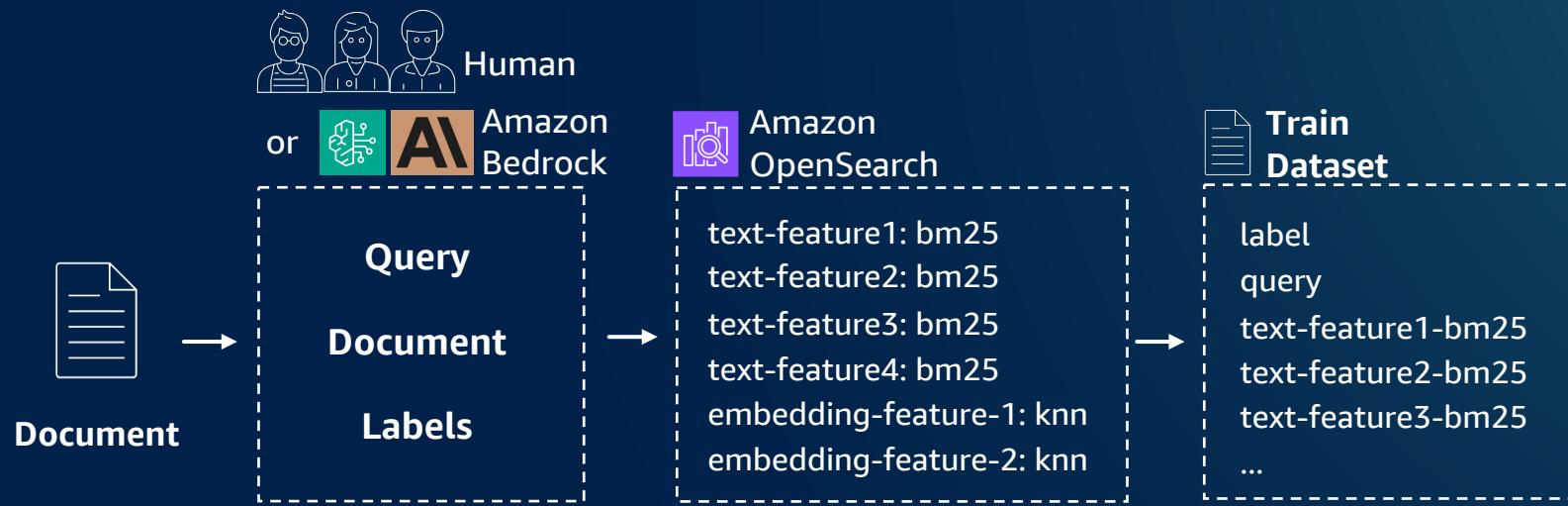
- Extract multiple text features and embedding features
- Utilize all features through a Ranking Model trained by LTR



https://en.wikipedia.org/wiki/Learning_to_rank

How to LTR (2/3)

Prepare Train Features

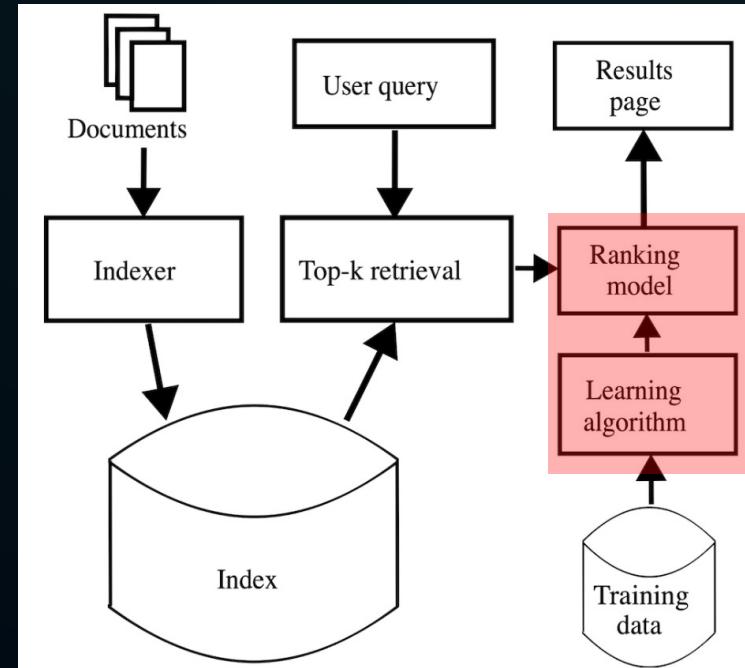


https://en.wikipedia.org/wiki/Learning_to_rank

- Generate LTR dataset from document features
- Use LLM for faster Query-Document-Label process

How to LTR (3/3)

Train and Upload LTR Model



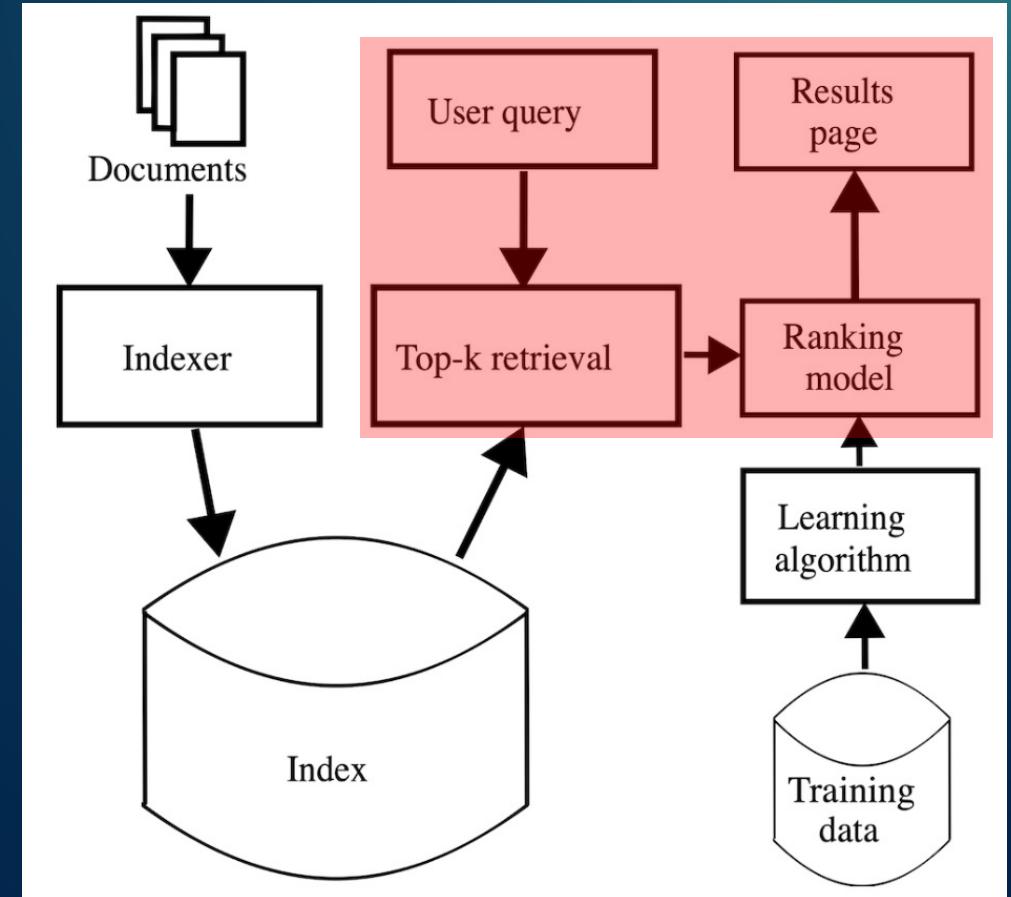
https://en.wikipedia.org/wiki/Learning_to_rank

- Train XGBoost model on prepared dataset
- Import model to OpenSearch as reranker

Let's search based on LTR Model

Building Search Query with LTR Model

```
GET /sm_semantic_demostore/_search
{
  "query": {
    // match fields..
  },
  "rescore": {
    "window_size": .. // window size,
    "query": {
      "rescore_query": {
        "sltr": {
          "params": {
            "feature_1": ...,
            "feature_2": ...
          },
          "model": "LTR_MODEL_NAME"
        }
      },
      "query_weight": 0.0, // relevance score weight
      "rescore_query_weight": 1.0 // LTR score weight
    }
  }
}
```



https://en.wikipedia.org/wiki/Learning_to_rank

Finally...

우리가 더 선호하는 결과를 기반으로, 원하는 코끼리 식당을 찾을 수 있습니다.



별점 1.4 (리뷰 7건)



별점 4.8 (리뷰 672건)

Part 2: 똑똑한 RAG를 '제대로' 운영하기

운영의 문제 정의

RAG의 서비스란..

RAG Service

모델 운영, 접근제어 운영, 문서 검토,
모델 평가 및 RAG 전체평가, 각종 CS 등



How engineer think AI



비즈니스로 확장되는 생성형 AI

2025년, 비즈니스 현장에서 실질적 가치를 만드는 생성형 AI 도입이 본격화되고 있습니다.

초기 검증 및 기능 확인

- 소규모 문서 기반 검토
- 검색 품질 및 서비스 효율성 검토
- 피드백 수집을 통한 이후 방향 설계

문서 확장 및 업무 적용

- 문서 확장으로 인한 검색 품질 하락
- 부서별 접근 제어 필요
- 보안 / 모니터링 등 운영 요건 증가

서비스 운영 및 비즈니스 성과

- 지속적인 문서 반영 필요
- 검색 품질 관련 운영 필요
- 지속적인 거버넌스 관리 필요

2024년

2025년, 올해

Part 2: 똑똑한 RAG를 '제대로' 운영하기

Index 운영 전략

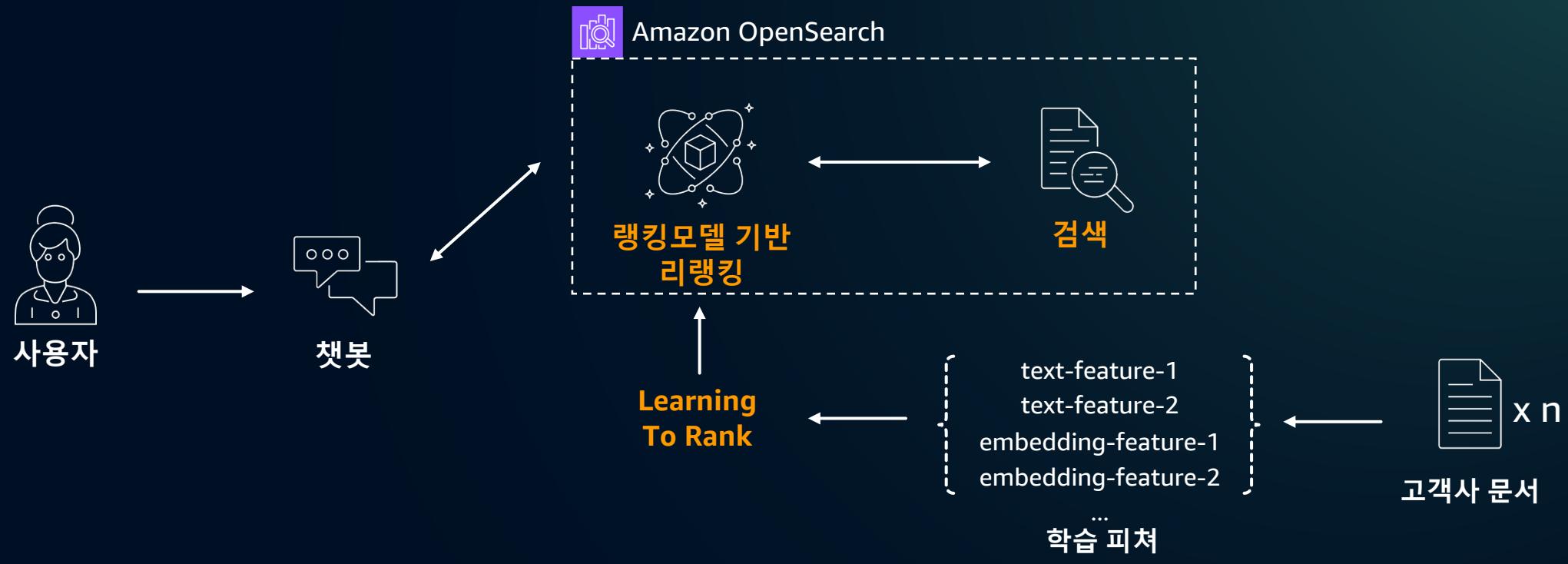
인덱싱 파이프라인



- 문서의 텍스트 정보 및 도표/이미지 정보등을 추출
- 추출한 여러 형태의 피쳐를 검색에 반영, 이를 기반하여 검색 활용

• 나아가, 하나 이상의 인덱스에 반영하는 요구사항이 올해 많이 발생하는 상황

검색 랭킹 혹은 Hybrid의 설정 또한 함께 필요



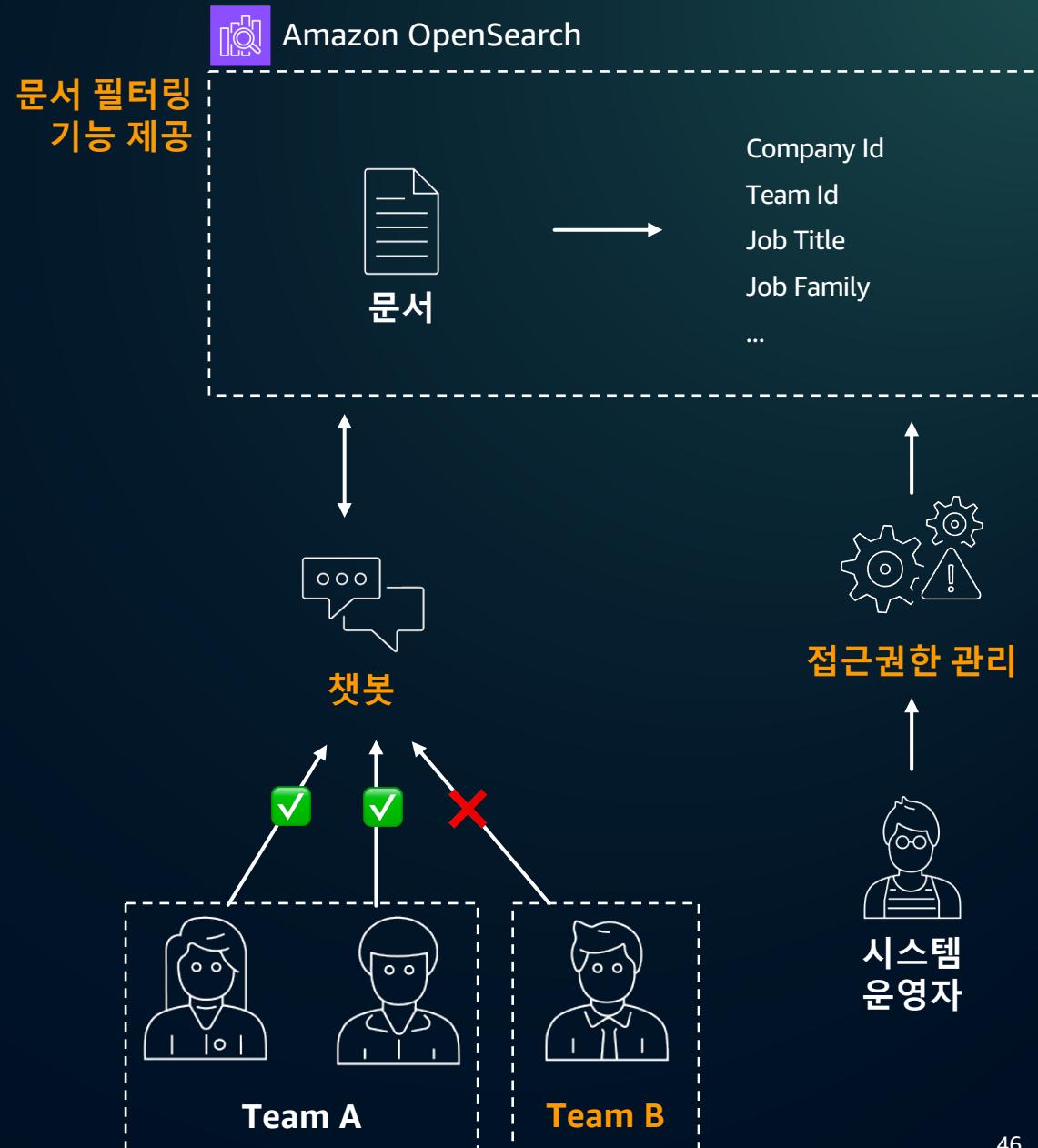
- 실제 고객사 문서의 특징들을 추출하고, Learning To Rank를 사용해 검색모델을 학습
- 생성된 랭킹 모델을 OpenSearch에 등록하여, 리랭킹 기능 제공

Part 2: 똑똑한 RAG를 '제대로' 운영하기

Governance

Governance

- 필요한 수준으로 문서 접근 거버넌스 설정
 - 예시: 회사, 팀, 역할 혹은 기술 태입 등으로 구분
- 개인/조직별 접근이 가능한 문서를 기반으로 답변 생성
- 접근 정보는 시스템 운영자가 설정하여 문서별로 부여하도록 함

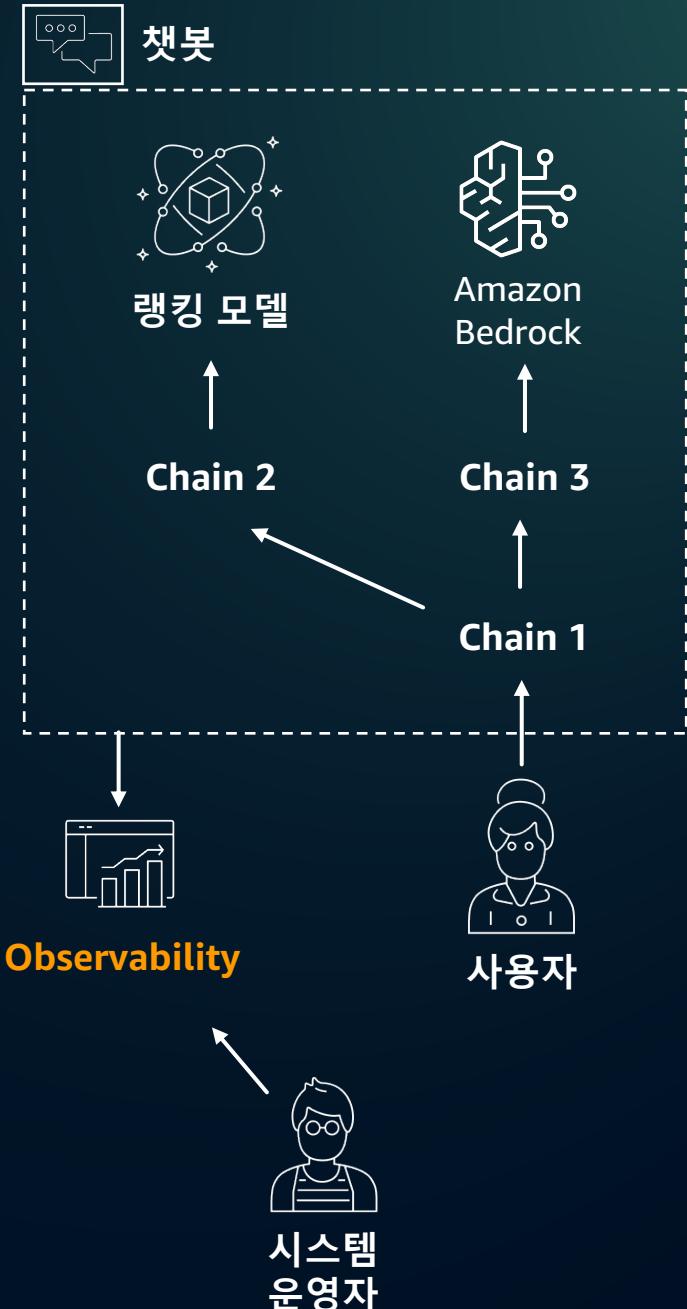


Part 2: 똑똑한 RAG를 '제대로' 운영하기

Observerability

Luga Chatbot : Observability

- 인덱스, 사전 모니터링 및 관리
- LLM 사용량, 비용 대시보드 제공
- LLM Chain 상세 정보:
 - 각 단계 입/출력
 - 실행 시간
 - 상세 프로파일링 가능



Luga UI : LLM Observability

The screenshot illustrates the Luga UI interface for LLM Observability, specifically focusing on a trace diagram and detailed log analysis.

Trace Details:

- Trace Status:** OK
- Latency:** 1.71s
- Evaluations:** Q&A Correctness incorrect, Hallucination hallucinated

Trace Diagram:

- query:** Latency 1.71s
- retrieve:** Latency 0.20s
- embedding:** Latency 0.12s
- synthesize:** Latency 1.51s
- llm:** Latency 1.51s, with a note: ⚡ 307

Logs and Evaluations:

- retriever:** retrieve
- Info:** My service is a hosting service designed for hosting your website. You can put your website on our se with accelerated CDN delivery, tracking of usage data for running your website. Our service is one of internet in terms of delivery and experience.
- Evaluations:** 0
- Attributes:** 0
- Events:** 0
- Input:** (Log content)
- Documents:** Relevance ndcg 0.00, Relevance precision 0.00, Relevance hit false
 - document d9027de8-b12c-4213-9486-eee18a2d44e0
- Evaluations:** Relevance irrelevant, score 0.00
 - The question is about a hosting service for websites with features like accelerated CDN delivery and usage data tracking. The referer other hand, talks about the deployment of Arize, a different service, in SaaS or managed deployment and On-Prem Deployment. It do any information about hosting websites, accelerated CDN delivery, or tracking usage data. Therefore, the reference text does not con that can help answer the question.

Searchdoc RAG Ops

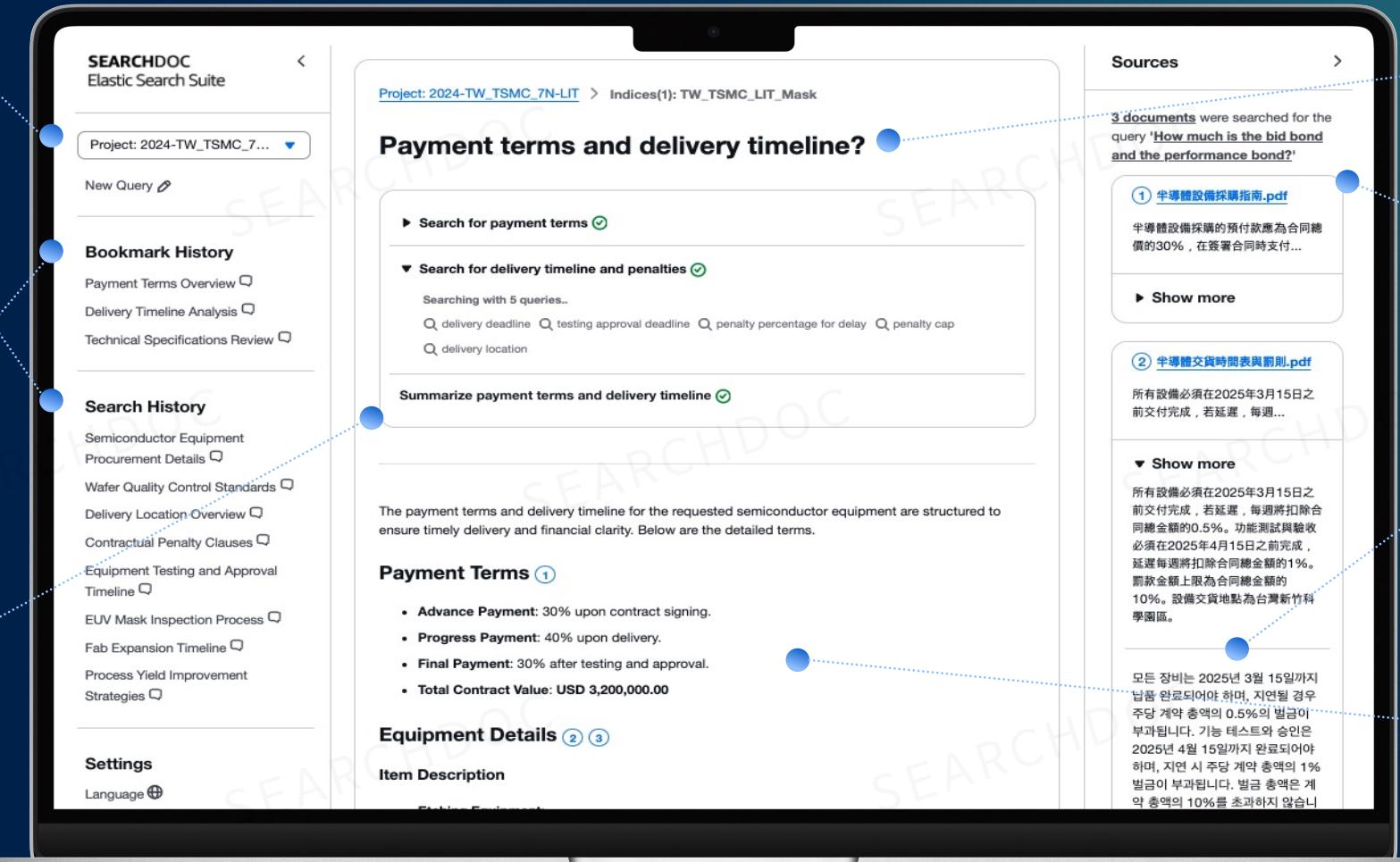
SEARCHDOC Elastic Search Suite

SEARCHDOC의 RagOps 솔루션, ESS(Elastic Search Suite) 대화형 검색서비스

검색 대상 프로젝트 선택

유저 별 검색 이력

검색 / 최종 보고서 생성간
상세 정보 제공



ESS Console : 유저 그룹 관리

RagOps의 시작은 사용자 관리로부터 시작합니다. 사용자와 그룹을 손쉽게 관리하고, 역할과 권한을 유연하게 제어할 수 있습니다.

The screenshot displays three main panels of the ESS Console:

- User Management Panel:** Shows a list of 8 users with columns for Name and Email. A search bar at the top allows finding users.
- Access Groups Management Panel:** Shows a list of 3 access groups with columns for Name and Description. A search bar at the top allows finding access groups.
- New Access Group Dialog:** A modal window titled "New Access Group" with fields for Name (Project AA Group), Description (For project AA, Temporary group), and Users (Select users). It lists users: empark, sarahlee1234, m.jung, dyoon, and olivia_han90.

ESS Console : 파일/인덱스 관리

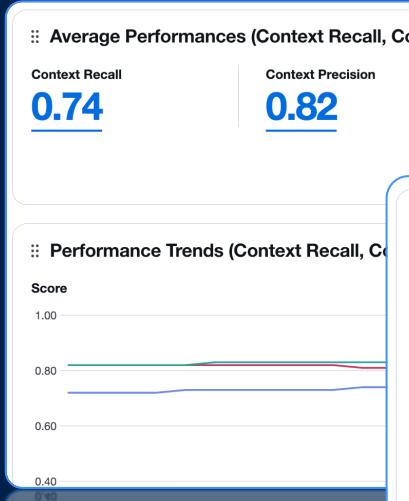
데이터셋과 애플리케이션을 한 곳에서 제어, 안정적인 운영과 신속한 디버깅을 지원합니다.

The screenshot displays the ESS Console interface with several panels:

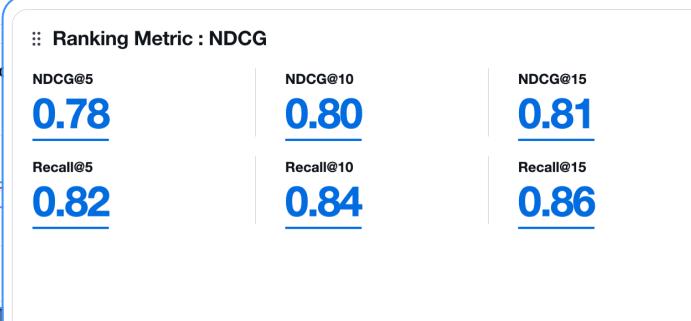
- Data Sources:** Shows three datasets: 2024-TW_TSMC_7N-LIT, 2024-KR_SSG_GAA, and 2024-US_INT_F42-EXP. Each dataset has a description, source storage (AWS S3), document count (150 for TW, 1676 for KR, 1017 for US), and an S3 path.
- Index Histories:** Displays a history of indexing operations for search indices, listing index names, imported documents, indexed documents, failed documents, and indexed chunks.
- Indices:** Shows a list of indices (5) with columns for index name, status, dictionary, data sources, running app, and daily index. One index, schdc-2024072001, is in progress (36% complete).
- File/DataSource Management:** A sidebar panel titled "파일/데이터소스 관리" (File/Data Source Management) with a description of the US INT F42 EXP project and source storage information.
- Index Details:** A detailed view of the "Pages - NDA_Template.docx" index, showing two chunks (Chunk 1 and Chunk 2) with their respective text content and ranking features (e.g., Chunk Text (Mandatory)).
- Indexing Features:** A panel titled "Chunk 1 - Details" showing ranking features like Chunk Text (mandatory), Section Title, and Keywords, along with an anchor text field.

ESS Console : 평가 및 모니터링

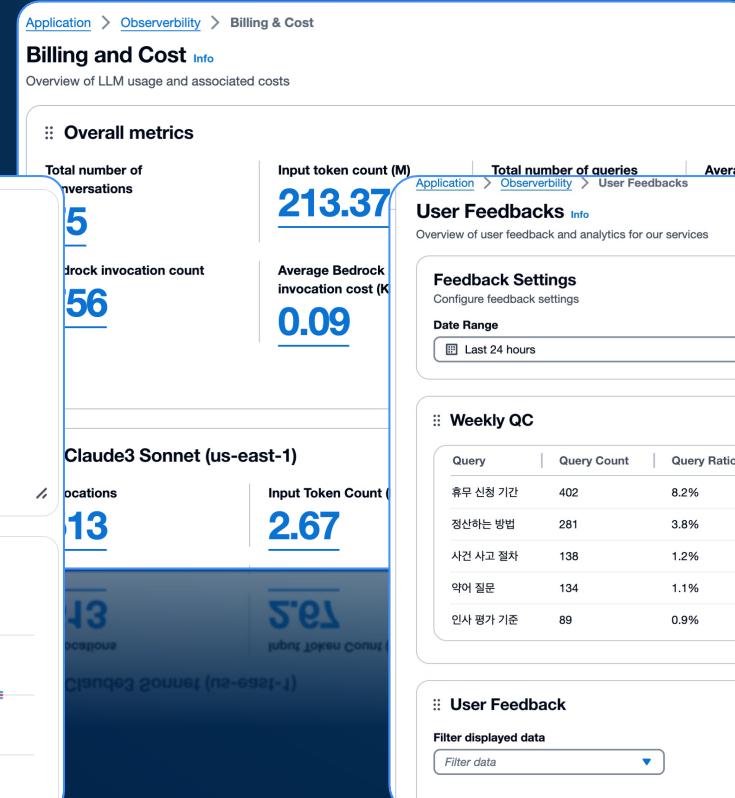
다양한 평가지표와 피드백을 실시간으로 확인, 데이터 기반 의사결정을 위한 대시보드를 제공합니다.



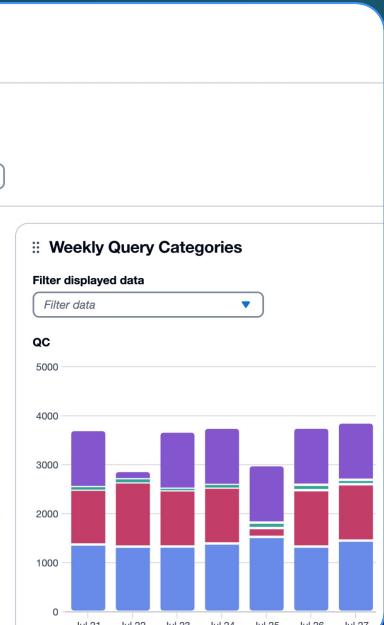
RAG 평가지표 관리



검색 관련 평가지표 관리



운영 관련 금액, 유저 대시보드 제공





SEARCHDOC

높은 수준의 검색 성능을 기반으로 한 대화형 검색 플랫폼

PoC 단계부터 서비스 운영까지, 다양한 산업 데이터를 효과적으로 검색하여
사용자에게 최적의 인사이트를 제공합니다



LinkedIn url

Thank you

감사합니다.