

강주란  
홍석범

# RAG 구성을 위한 ElasticSearch 활용

Retrieval  
Augmentation  
Generation

# INDEX

## 목차

### 1. RAG와 ElasticSearch

A. RAG를 위한 Database

### 2. 텍스트 기반 검색

A. 색인

B. 쿼리

### 3. Vector 검색

A. BOW에서 Embedding으로

B. ANN

### 4. 하이브리드 검색

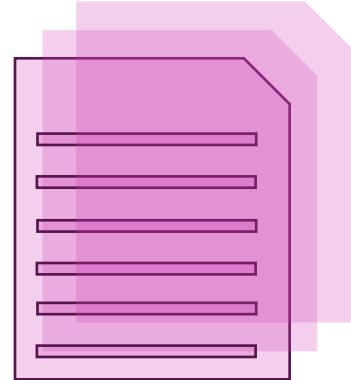
A. Text와 Vector의 조합

# RAG and Elasticsearch

## RAG와 ElasticSearch

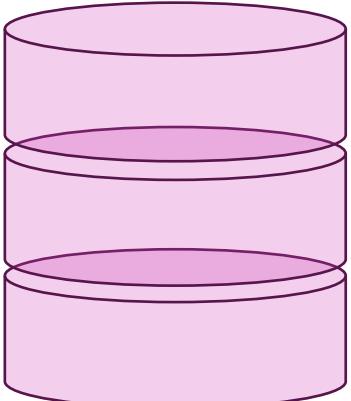
### RAG 를 위한 Database

데이터 임베딩 및  
벡터 DB 구성



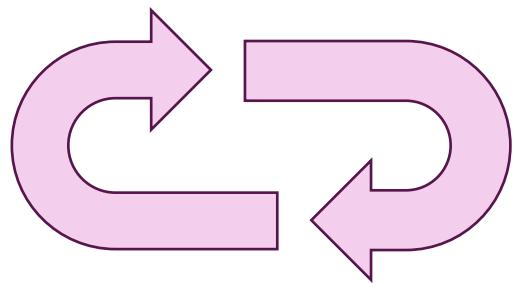
Retrieval

쿼리 벡터화 및  
관련 정보 추출  
(증강 단계)



Augmented

LLM을 통한  
답변 생성



Generation

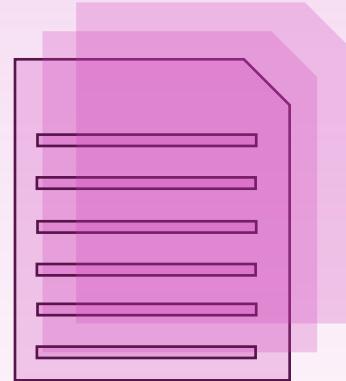
# RAG and Elasticsearch

## RAG와 ElasticSearch

### RAG 를 위한 Database

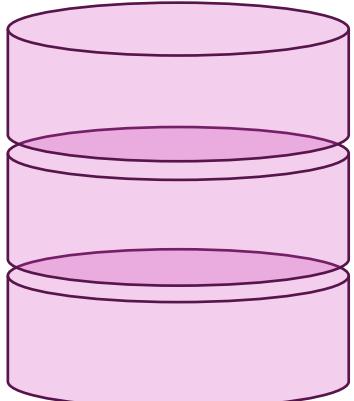
#### < ElasticSearch 활용 >

데이터 임베딩 및  
벡터 DB 구성



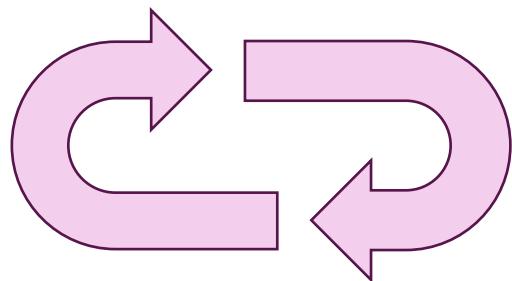
Retrieval

쿼리 벡터화 및  
관련 정보 추출  
(증강 단계)



Augmented

LLM을 통한  
답변 생성



Generation

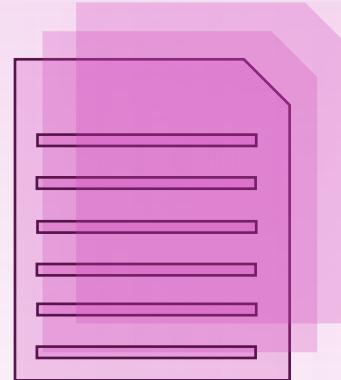
# RAG and Elasticsearch

## RAG와 ElasticSearch

### RAG 를 위한 Database

#### < ElasticSearch 활용 >

데이터 임베딩 및  
벡터 DB 구성



## Retrieval

색인

벡터를 주어진 데이터 구조에 매핑

쿼리

인덱스 벡터를 쿼리 벡터와  
비교하여 최근접 벡터 항목을 결정

# Text Based Search

Text 기반 검색

## 목차

### A. 색인

1) 역인덱스

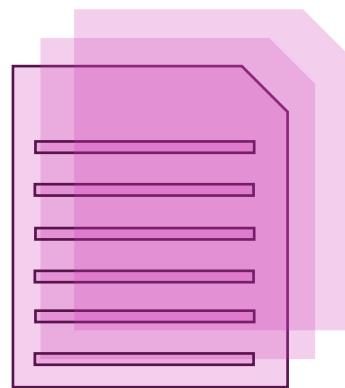
2) analyzer

### B. 쿼리

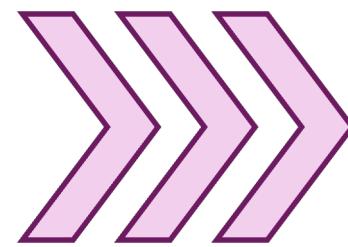
INDEX

# Text 기반 검색 Based Search

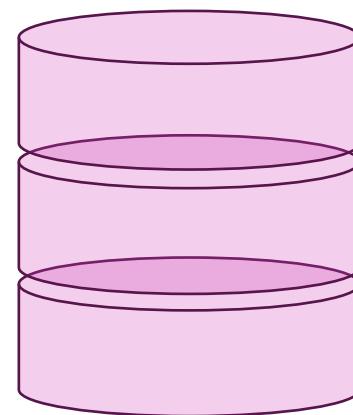
## RAG 를 위한 Database



Raw Data



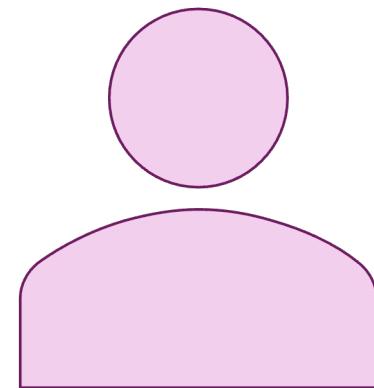
Indexing



Index



Search



User

# Text 기반 검색 Based Search

## 색인, Tokenizer and Analyzer

### indexing 색인

데이터가 검색될 수 있는 구조로 변경하기 위해  
원본 문서를 검색어 토큰으로 변환하여 저장하는 일련의 과정

Reverse  
Indexing

text analysis

Analyzer

Char Filters

Tokenizer

Token Filters

# Text 기반 검색 Based Search

## 색인, Tokenizer and Analyzer

indexing  
색인

데이터가 검색될 수 있는 구조로 변경하기 위해  
원본 문서를 검색어 토큰으로 변환하여 저장하는 일련의 과정

Reverse  
Indexing

text analysis

Analyzer

Char Filters

Tokenizer

Token Filters

# Text Based Search

Text 기반 검색

역 인덱싱

ID	Content
1	The quick brown fox
2	The quick brown fox jumps over the lazy dog
3	The quick brown fox jumps over the quick dog
4	Brown fox brown dog
5	Lazy jumping dog

# Text 기반 검색 Based Search

## Query 쿼리

데이터베이스나 데이터 Repository 시스템에서  
데이터나 정보를 요청하는 것

**TF-IDF** 단어 빈도와 문서의 역문서 빈도를 기반으로 각 단어의 중요도를 평가합니다.

**BM25** 문서 길이와 같은 추가 요소를 사용해 TF-IDF 를 보완한 알고리즘 입니다.

# Text 기반 검색 Based Search

## Query 쿼리

데이터베이스나 데이터 Repository 시스템에서  
데이터나 정보를 요청하는 것

**TF-IDF** 단어 빈도와 문서의 역문서 빈도를 기반으로 각 단어의 중요도를 평가합니다.

**BM25** 문서 길이와 같은 추가 요소를 사용해 TF-IDF 를 보완한 알고리즘 입니다.

# Vector Based Search

## Vector 기반 검색

### 목차

A. BOW에서 Embedding으로

B. 거리를 측정하는 계산식

- 1) Cosine Similarity
- 2) Dot Product Similarity
- 3) Euclidean Distance

C. ANN

- 1) LSH
- 2) KD Trees
- 3) Annoy

I N D E X

# Vector Based Search

## Vector 기반 검색

### TEXT 기반 검색의 개념

- 키워드 매칭에 의존합니다.
- 쿼리와 일치하는 키워드나 구문을 검색 결과로 반환합니다.

### TEXT 기반 검색 :: 한계점

- 텍스트의 문맥이나 의미를 반영할 수 없습니다.
- 문서의 내용과 검색 의도 간의 차이가 있을 수 있습니다.

### VECTOR 기반 검색의 개념

- 벡터 기반 검색은 텍스트나 이미지를 고차원 벡터로 변환하여 의미적 유사성을 평가하는 방식입니다.
- 문맥과 의미를 더 잘 반영한 검색이 가능합니다.
- 자연어 처리(NLP), 이미지 검색, 추천 시스템 등 다양한 응용 분야에서 활용됩니다.

# Vector Based Search

## BoW에서 Embedding으로

### Bag of Words

텍스트를 단어의 출현 빈도로 표현하는 방법입니다.

단어의 출현율이 서로 다른 문서 사이에서 유사하다면,  
비슷한 문서라고 판단합니다.

BoW 모델은 단어의 순서나 문맥을  
반영하지 않기 때문에,  
의미적 유사성을 포착하지 못할 수 있습니다.

#### Raw Text

it is a puppy and it  
is extremely cute

#### Equivalent BoW vectors

it	2
they	0
puppy	1
and	1
cat	0
aardvark	0
cute	1
extremely	1
...	...

≡

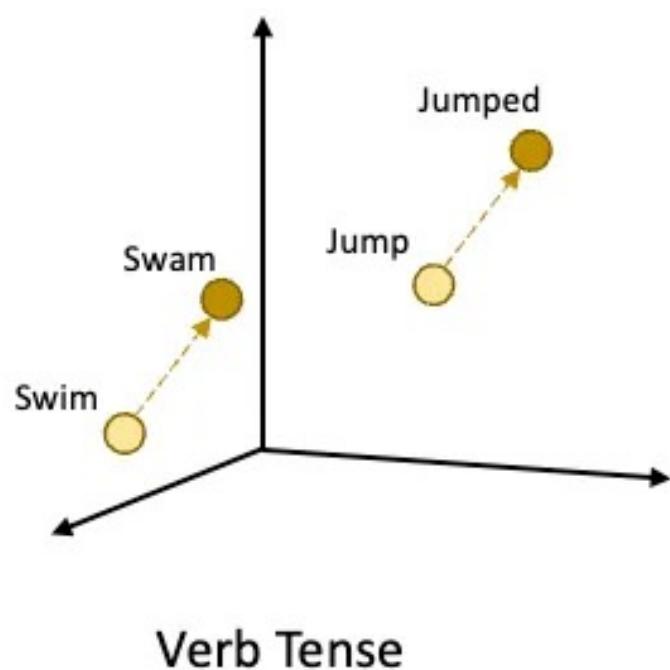
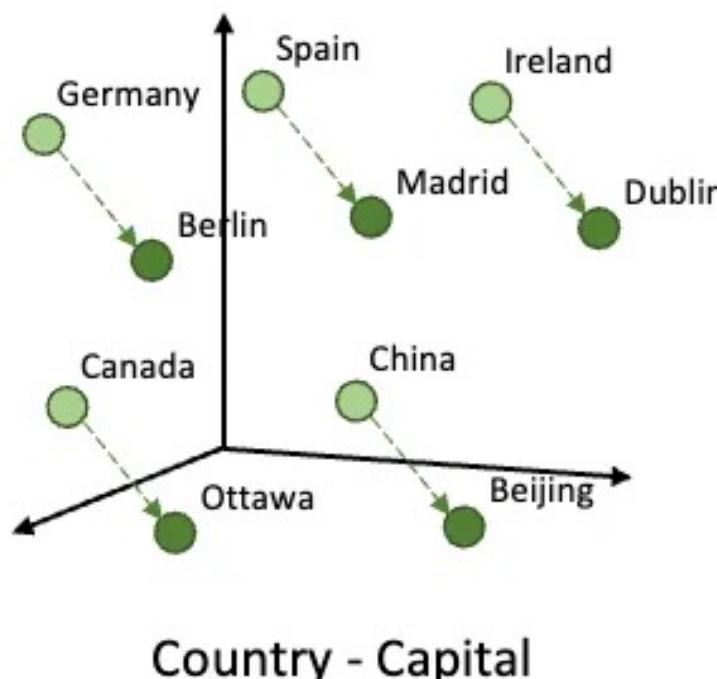
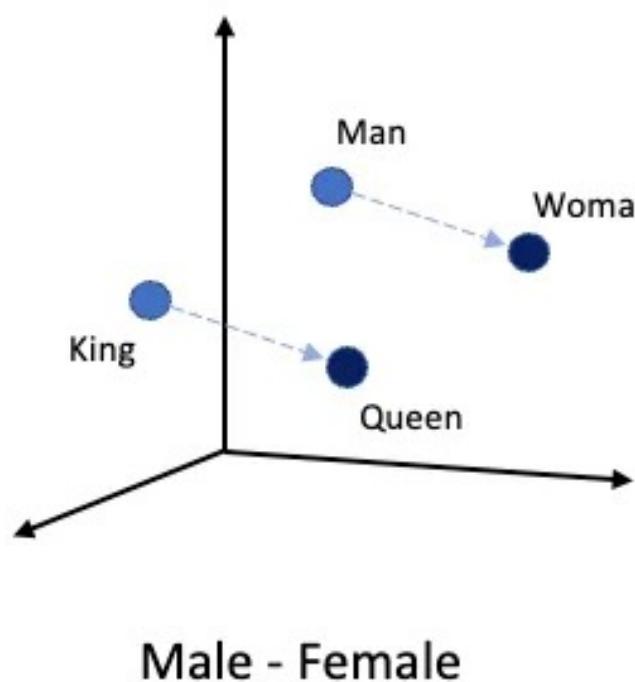
it	2
they	0
puppy	1
and	1
cat	0
aardvark	0
cute	1
extremely	1
...	...

# Vector Based Search

## Vector 기반 검색 BoW에서 Embedding으로

### Embedding

단어와 문장을 고차원 벡터 공간에 매핑하여 의미를 반영한 벡터를 생성합니다.



# Vector Based Search

## Vector 기반 검색 BoW에서 Embedding으로

### Embedding

단어와 문장을 고차원 벡터 공간에 매핑하여 의미를 반영한 벡터를 생성합니다.

**Embedding 모델은 문맥과 의미를 반영하기 때문에, BoW 보다 텍스트 간의 의미적 유사성을 더 정확하게 측정할 수 있습니다.**

**BoW는 단순하고 직관적이지만, 의미적 유사성 측면에서 Embedding 모델이 더 강력합니다.**

Male - Female

Country - Capital

Verb Tense

# Vector Based Search

## Vector 기반 검색

거리를 측정하는 계산식

코사인 유사성

**Cosine Similarity**

두 벡터 간의 각도를 측정

텍스트 임베딩에서  
두 문장의 의미적 유사성을 비교

점 곱 유사성

**Dot Product Similarity**

크기와 방향에 따라 유사성을 측정

추천 시스템에서  
사용자가 선호하는 항목과  
유사한 항목 찾기

유클리드 거리

**Euclidean Distance**

두 벡터 간의 직선 거리를 측정

이미지 검색에서  
피처 벡터 간의  
시각적 유사성을 평가

각 Vector 간의 유사도를 정확하게 계산 가능하지만,  
반대로 계산에 걸리는 시간이 긴 단점도 있습니다.

# Vector Based Search

## Approximate Nearest Neighbor

### 정확한 거리 측정의 문제점

고차원 벡터 공간에서는 모든 데이터 포인트 간의 거리를 계산하는 것이 비효율적입니다.  
연산 시간이 오래 걸리고, 대규모 데이터셋에서는 현실적으로 불가능할 수 있습니다.

“ ”  
근사 최근접 이웃(ANN) 검색은 정확성을 일부 포기하는 대신,  
빠르고 효율적으로 유사한 데이터를 찾는 방법입니다.

# Vector Based Search

## Approximate Nearest Neighbor

### 1 지역성 기반 해싱 (LSH)

유사한 벡터들이 동일한 해시 버킷에 맵핑되도록 설계된 해싱 기법입니다.

### 2 KD 트리

차원별로 데이터를 분할하여 트리 구조로 저장하는 방법입니다.  
저차원 데이터에서 효율적이며, 트리를 탐색하여 근사 최근접 이웃을 찾습니다.

### 3 Annoy

여러 개의 랜덤 KD 트리를 생성하여, 근사 최근접 이웃을 효율적으로 찾는 방법입니다.  
트리 탐색 결과를 결합하여 유사한 데이터를 빠르게 검색할 수 있습니다.

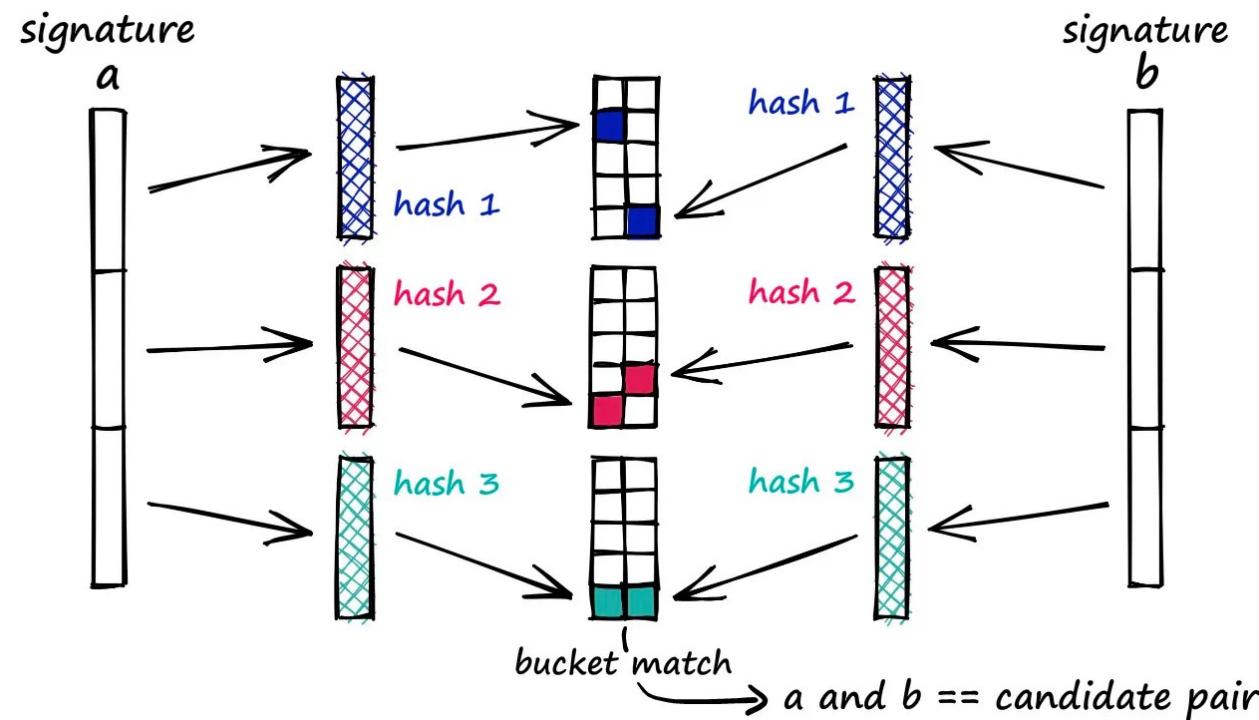
# Vector Based Search

## Vector 기반 검색 Approximate Nearest Neighbor

### 1 지역성 기반 해싱 (LSH)

유사한 벡터들이 동일한 해시 버킷에 매핑되도록 설계된 해싱 기법입니다.

3  
Ani  
여러 기  
트리 트



방법입니다.  
1사 최근접 이웃을 찾습니다.

찾는 방법입니다.  
다.

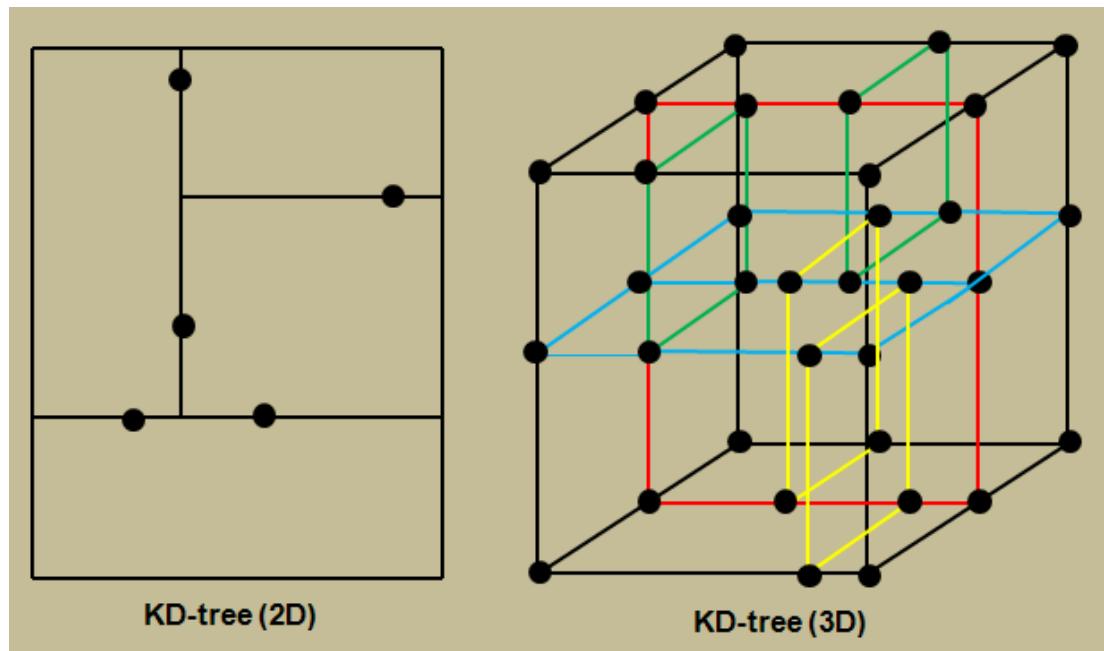
# Vector Based Search

## Vector 기반 검색 Approximate Nearest Neighbor

### 2 KD 트리

차원별로 데이터를 분할하여 트리 구조로 저장하는 방법입니다.

저차원 데이터에서 효율적이며, 트리를 탐색하여 근사 최근접 이웃을 찾습니다.



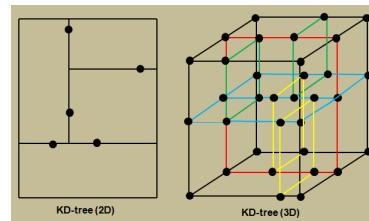
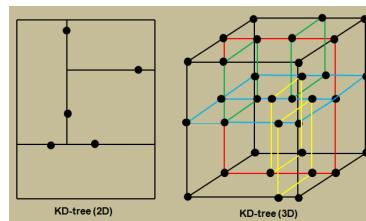
KD 트리는 차원이 증가할수록  
성능이 급격히 저하되기 때문에,  
고차원 데이터에는 적합하지 않을 수 있습니다.

# Vector Based Search

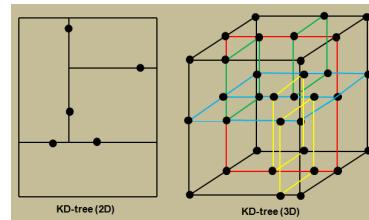
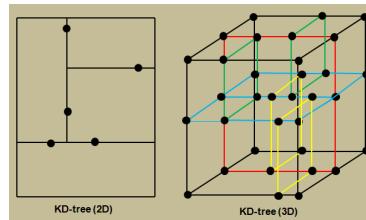
## Approximate Nearest Neighbor

### 3 Annoy

여러 개의 랜덤 KD 트리를 생성하여, 근사 최근점 이웃을 효율적으로 찾는 방법입니다.  
트리 탐색 결과를 결합하여 유사한 데이터를 빠르게 검색할 수 있습니다.



여러 개의 랜덤 KD 트리를 생성하여,  
근사 최근점 이웃을 효율적으로 찾는 방법입니다.

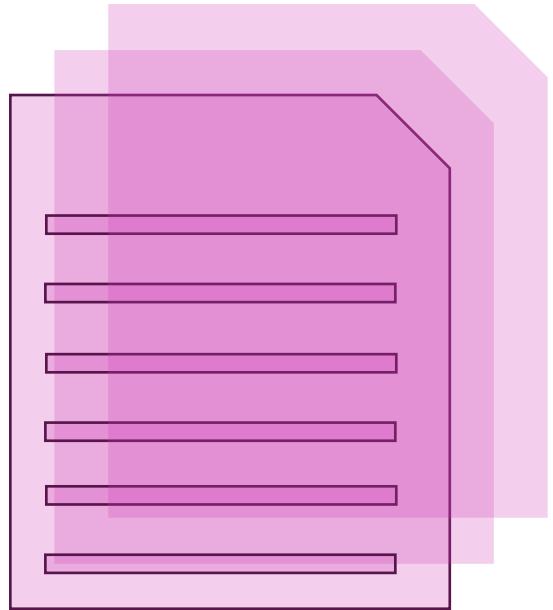


트리 탐색 결과를 결합하여 빠르게 검색할 수 있습니다.

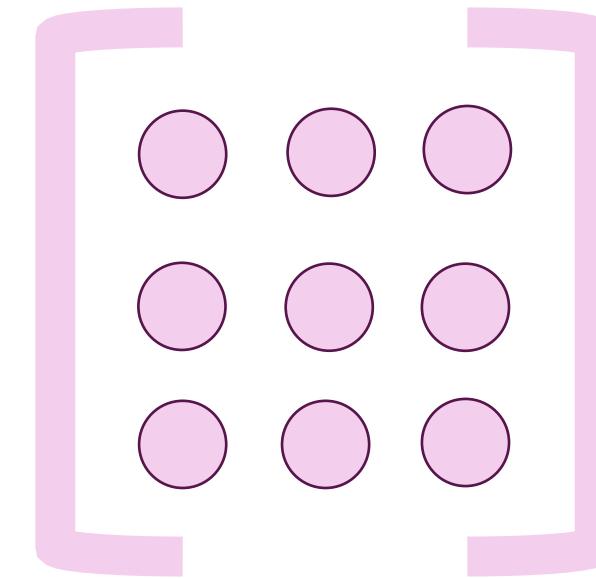
# Hybrid Based Search

Hybrid 검색

TEXT 기반 검색과 Vector 기반 검색 결과를 조합



TEXT

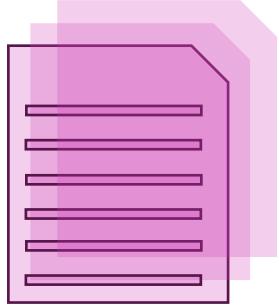


Vector

# Hybrid 검색

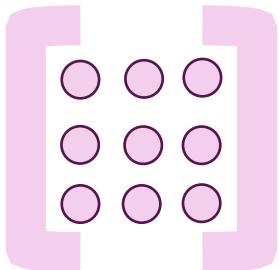
# Based Search

TEXT 기반 검색과 Vector 기반 검색 결과를 조합



## TEXT

- 빠른 결과 탐색
- 동의어, 유의어 사전을 통한 토큰화



## Vector

- 문맥을 반영한 탐색
- 고차원 벡터 정보로 변환

## Scoring

전통적인 텍스트 기반 검색 결과와  
Vector 기반의 유사도 계산 결과를  
점수로 수치화 합니다.

수치화 된 점수를 통해  
각자에게 가장 잘 맞는 방식으로  
조정 가능합니다.

# Questions Answers



# REFERENCE 출처

- Deep Learning Bible - Natural Language Processing
- The Power of Embeddings in Machine Learning - Rian Dolphin
- Faiss: The Missing Manual - James Briggs
- Voxel-based volume modelling of individual trees using terrestrial laser scanners - John C. Trinder
- what-is query-language - elastic.co

# REFERENCE 출처

- Deep Learning Bible - Natural Language Processing
- Understanding retrieval augmented generation (RAG)  
Elastic Snackable Series
- What is retrieval augmented generation – elastic.co
- Neural networks Series - 3Blue1Brown
- Superb\_AI Vector Store Explained - Superb\_AI