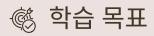


Spring AI 활용 - RAG 구현

임베딩과 벡터 저장소의 원리를 이해하고, 지능형 RAG와 질문 변환 기술을 활용하여 도메인 특화 챗봇을 구현하는 실전 가이드입니다.





학습 목표

원리 이해

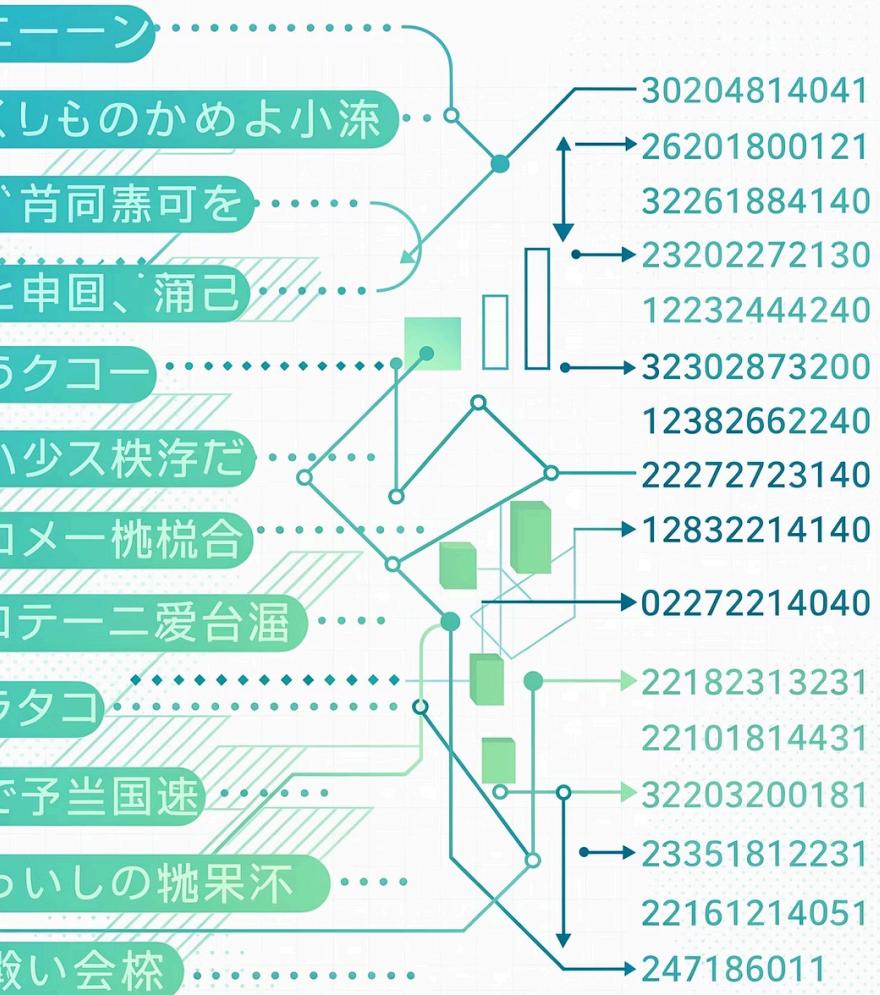
임베딩과 벡터 저장소의 핵심 메커니즘을 이해하고, 지능형 RAG(Advisors)와 질문 변환(Query Engineering)의 작동 방식을 설명할 수 있습니다.

실전 구현

Spring AI를 활용해 다양한 포맷의 데이터를 정제하여 벡터 DB에 적재하고, 대화 맥락이 유지되는 도메인 특화 챗봇을 설계 및 구현할 수 있습니다.

품질 향상

사용자의 복잡한 질문 의도를 정확히 파악하기 위해 지속적으로 쿼리를 확장 및 개선하고, 데이터의 정밀한 엔지니어링을 통해 답변의 신뢰성을 확보하는 자세를 갖출 수 있습니다.



임베딩과 벡터 저장소

임베딩은 텍스트, 이미지, 오디오와 같은 비정형 데이터를 컴퓨터가 이해할 수 있는 **부동 소수점 숫자의 배열(벡터)**로 변환하는 과정입니다.

임베딩 모델

이 변환 작업을 수행하는 AI 모델입니다. 모델에 따라 수백에서 수천 차원의 벡터로 출력됩니다.

왜 벡터로 변환하는가?

데이터를 다차원 공간상의 한 점으로 표현하면, **수학적인 거리 계산**을 통해 데이터 간의 의미적 유사성을 파악할 수 있기 때문입니다.

벡터 저장소의 작동 원리

데이터 입력

임베딩 모델

벡터 변환

저장소에 저장

임베딩 모델이 생성한 벡터는 벡터 저장소에 저장되며, 이곳에서 효율적인 검색이 이루어집니다. 의미가 유사한 데이터끼리 군집을 이루어 저장됩니다.

유사도 높음

벡터의 방향과 크기가 비슷함 (예: "강아지"와 "애완견")

유사도 낮음

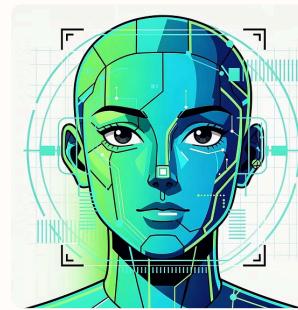
벡터의 방향과 크기가 크게 다름 (예: "강아지"와 "물고기")

실무 활용 사례



문서 검색 (RAG)

"대통령 임기"에 대해 질문하면, 헌법 전문 중 임기 관련 문장(유사 벡터)을 찾아내어 답변에 활용합니다.



안면 인식

카메라에 찍힌 얼굴을 벡터화하여 저장된 데이터와 비교함으로써 누군지 식별합니다. 이를 LLM과 결합하면 개인화된 대화가 가능해집니다.

- 주의사항:** 저장할 때 사용한 임베딩 모델과 검색할 때 사용하는 모델은 반드시 동일해야 합니다. 서로 다른 모델을 사용하면 검색 결과가 부정확해집니다.

PGVector 설치 및 설정

PGVector란?

PostgreSQL의 오픈 소스 확장 기능으로, 벡터 데이터를 컬럼 타입으로 지원합니다.

주요 장점

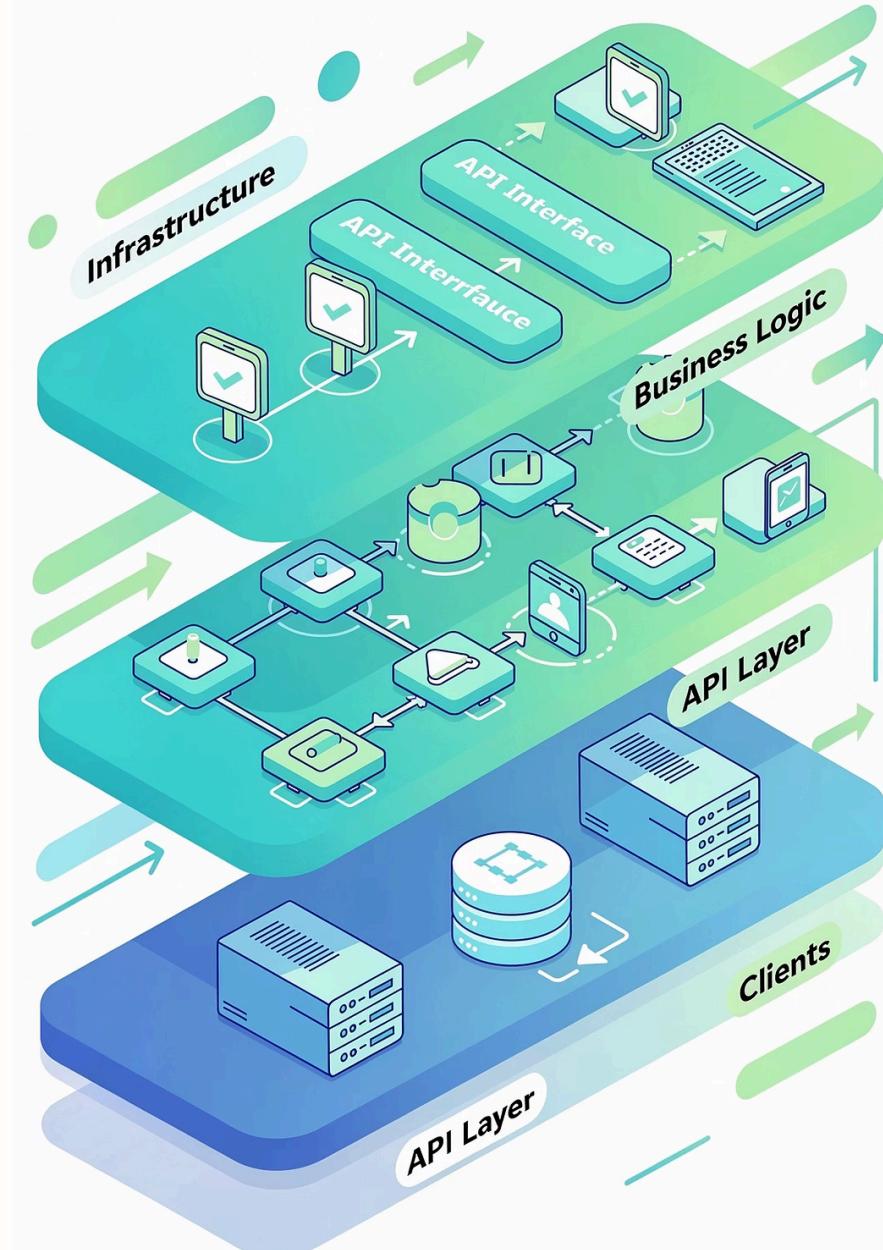
- 기존 관계형 데이터와 벡터 데이터를 한 곳에서 통합 관리
- SQL 쿼리 및 PostgreSQL의 기존 인덱싱 기능 활용 가능
- Spring AI의 VectorStore 인터페이스와 쉽게 연동

Docker를 통해 간편하게 설치할 수 있으며, Spring Boot 3.5.x와 Spring AI 1.1.2 버전의 조합으로 안정적인 환경을 구축할 수 있습니다.

Spring AI Embedding API 구조

Spring AI의 Embedding API는 입력(EmbeddingRequest) → 처리(EmbeddingModel)
→ 결과(EmbeddingResponse)로 이루어지는 표준 파이프라인을 구축합니다.

- 1 Embedding Model Implementations
OpenAI, Mistral, Vertex AI 등 다양한 프로바이더 지원
- 2 Embedding Model API
임베딩 서비스의 핵심 추상화 계층
- 3 Generic Model API
Spring AI의 기본 모델 추상화 계층



OpenAI 텍스트 임베딩 모델

모델	벡터차원	최대 토큰	MIRACL	MTEB	특징
text-embedding-ada-002	1536	8192	31.4%	61.0%	2세대 모델, 안정적 성능
text-embedding-3-small	1536	8192	44.0%	62.3%	3세대, 다국어 성능 향상
text-embedding-3-large	3072	8192	54.9%	64.6%	최대 차원, 최고 성능

Spring AI 1.1.2 버전은 기본적으로 text-embedding-ada-002 모델을 사용하며, application.yaml 설정을 통해 다른 모델로 변경할 수 있습니다.



Document 저장 및 검색

Spring AI에서 Document는 콘텐츠(Text/Media)와 메타데이터를 하나로 묶는 컨테이너입니다. VectorStore.add() 메서드는 Document 객체 목록을 받아 임베딩을 수행하고 벡터 저장소에 저장합니다.

id

Document를 식별하는
UUID

text/media

실제 임베딩될 콘텐츠

metadata

출처, 날짜 등 부가 정보

score

유사도 검색 결과 점수

RAG (Retrieval-Augmented Generation)



RAG는 LLM의 한계를 극복하기 위해 외부 지식을 실시간으로 참조하는 방식입니다.

01

Retrieve (검색)

사용자 질문을 벡터로 변환하여 Vector Database에서 관련 문서를 검색합니다.

02

Augment (증강)

검색된 정보를 사용자 질문과 합쳐서 프롬프트를 재구성합니다.

03

Generate (생성)

LLM이 증강된 프롬프트를 바탕으로 최종 답변을 생성합니다.

EXTRACT

TRANSFORM

LOAD



ETL 파이프라인

Spring AI는 외부 데이터를 지식 기반 저장소로 만드는 과정을 **ETL(Extract, Transform, Load)** 구조로 정형화합니다.

1

Extract (추출)

DocumentReader를 사용하여 PDF, Word, URL 등 다양한 소스에서 텍스트를 추출합니다.

2

Transform (변환)

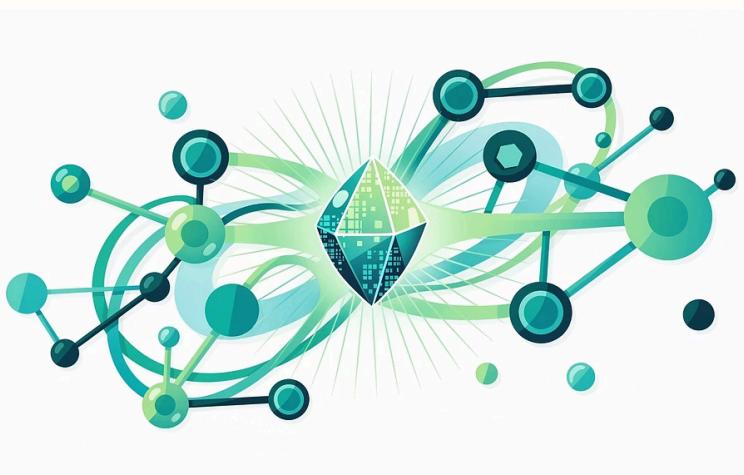
TokenTextSplitter로 문서를 작은 조각으로 분할하고, KeywordMetadataEnricher로 키워드를 추가합니다.

3

Load (적재)

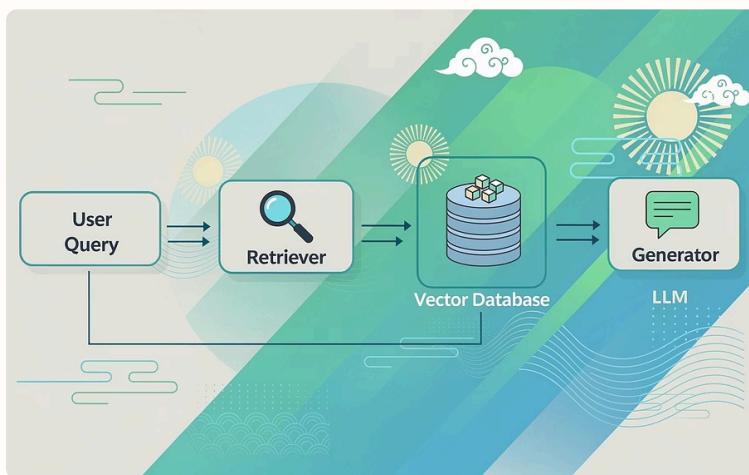
DocumentWriter를 통해 변환된 Document 조각들을 벡터 저장소에 최종 적재합니다.

지능형 RAG: Spring AI Advisors



QuestionAnswerAdvisor

RAG 패턴을 가장 쉽게 구현할 수 있는 Advisor로, 사용자 질문을 기반으로 벡터 저장소에서 관련 문서를 자동으로 검색하고 프롬프트에 포함시킵니다.



RetrievalAugmentationAdvisor

RAG의 각 단계(검색, 증강, 생성)를 모듈화하여 설계된 Advisor로, 런타임 시 유연하게 조합 가능합니다.

검색 전(Pre-Retrieval) 모듈



CompressionQueryTransformer

대화 맥락을 파악하여 모호한 질문을 명확하고 완전한 문장으로 변환합니다.



RewriteQueryTransformer

질문 내 불필요한 노이즈를 제거하여 검색 효율을 높이도록 재구성합니다.



TranslationQueryTransformer

질문을 임베딩 모델이 최적으로 인식하는 대상 언어로 번역합니다.



MultiQueryExpander

하나의 질문을 여러 개의 유사 질문으로 확장하여 검색 범위를 넓힙니다.

Query Transformer 비교

CompressionQueryTransformer

입력: "국회의원은?"

출력: "국회의원의 임기는 몇 년입니까?"

이전 대화 맥락을 참조하여 불완전한 질문을 완전한 문장으로 변환합니다.

RewriteQueryTransformer

입력: "국회의원은 하는 일 없이 당파 싸움만..."

출력: "국회의원의 의무와 역할에 대해 알려주세요."

감정적 표현과 불필요한 서술을 제거하고 핵심만 추출합니다.

TranslationQueryTransformer

입력: "Wie lange dauert die Amtszeit?"

출력: "대통령의 임기는 얼마나 됩니까?"

다국어 질문을 대상 언어로 번역하여 검색 정확도를 높입니다.

MultiQueryExpander

입력: "대통령의 임기는?"

출력: 3개의 유사 질문 생성

단일 질문을 다양한 관점의 질문으로 확장하여 검색 범위를 넓힙니다.

핵심 요약 및 다음 단계

임베딩 & 벡터 저장소

텍스트를 벡터로 변환하여 의미적 유사도 검색을 가능하게 합니다.

RAG 파이프라인

검색-증강-생성의 3단계로 LLM의 한계를 극복합니다.

ETL 프로세스

다양한 포맷의 데이터를 추출-변환-적재하여 지식 베이스를 구축합니다.

지능형 Advisors

Query Transformer를 활용하여 검색 품질을 극대화합니다.

이제 학습한 내용을 바탕으로 도메인 특화 챗봇을 직접 구현해보세요. Spring AI의 강력한 기능들을 활용하여 정확하고 신뢰할 수 있는 AI 서비스를 만들 수 있습니다.

