벡터스토어: RAG어플리케이션

❑ LLM의 한계를 뛰어 넘는 방식

전용 LLM 챗봇의 필요성: LLM의 한계를 뛰어넘는 방식

- **Finetuning (파인튜닝)** : 기존 LLM에 특정 도메인 데이터를 추가 학습하여 모델을 최적화하는 방법
- **RAG (검색 증강 생성):** 외부 지식 소스와 연계하여 범용성과 적응력을 유지하면서도 정확하고 신뢰할 수 있는 답변을 생성하는 방법

파인튜닝(Fine-Tuning,미세조정)

- 이미 학습된 모델에 새로운 데이터를 추가로 학습시켜 성능 향상
- 기존에 잘 훈련된 모델의 파라미터를 조정하여 새로운 데이터에 맞게 최적화
- 예:기본적으로 수영을 잘하는 사람에게 새로운 수영 기술을 가르쳐 그 기술에 맞춰 몸의 움직임을 세밀하게 조정하는 것

벡터스토어: RAG어플리케이션

❑ LLM의 한계를 뛰어 넘는 방식

파인튜닝의 장점

- 사전 학습된 모델을 기반으로 하므로, 처음부터 학습할 필요 없이 기존의 지식을 활용하여 효율적으로 모델을 개선할 수 있음.
- 특정 작업이나 도메인에 맞춰 모델을 최적화하여 높은 정확도와 성능을 발휘하고, 맞춤형 솔루션을 제공

파인튜닝의 한계

- **고비용 및 시간 소요**: 대규모 데이터셋을 사용해 모델을 재훈련해야 하므로 **많은 시간과 비용**이 듦
- **지속적 유지보수 필요**:데이터와 요구사항이 변할 때마다 모델을 다시 파인튜닝해야 하므로, **지속적인 유지보수**가 필요
- **유연성 부족**:새로운 정보나 상황에 적응하기 어려워 다양한 상황에서의 적용 가능성이 제한

-> 비즈니스에서 가장 중요한 시간 과 비용의 문제

		Time (GPU hours)	Power Consumption (W)	Carbon Emitted (tCO_2eq)
Llama 2	7B	184320	400	31.22
	13B	368640	400	62.44
	34B	1038336	350	153.90
	70B	1720320	400	291.42
Total		3311616		539.00

벡터스토어: RAG어플리케이션

❑ LLM의 한계를 뛰어 넘는 방식

RAG(검색증강 생성)

- 외부 지식 소스와 연계하여 범용성과 적응력을 유지하면서도 정확하고 신뢰할 수 있는 답변을 생성하는 방법



벡터스토어: RAG어플리케이션

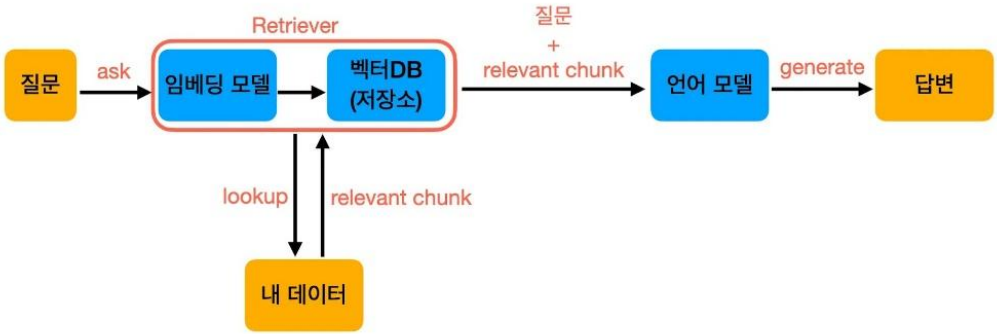
❑ LLM의 한계를 뛰어 넘는 방식

RAG(검색증강 생성)

- 외부 지식 소스와 연계하여 정확하고 신뢰할 수 있는 답변을 생성하는 기술

RAG(검색증강 생성)의 장점

- **프롬프트 비용 절감**: 외부 데이터를 별도 검색 시스템에 저장하여 시간,비용 절약
- **효율성:** 파인튜닝(미세 조정) 없이 외부 데이터를 반영하여 파인튜닝에 비해 시간과 비용이 적게 듬.
- **신뢰도 향상**: 기본 LLM보다 신뢰할 수 있는 내용을 생성하며, 할루시네이션 가능성을 줄이고, 답변의 근거를 제시가능
- **문서 업데이트 및 최선 정보 반영 용이:** 실시간으로 필요한 정보를 검색하여 최신 정보 반영 가능

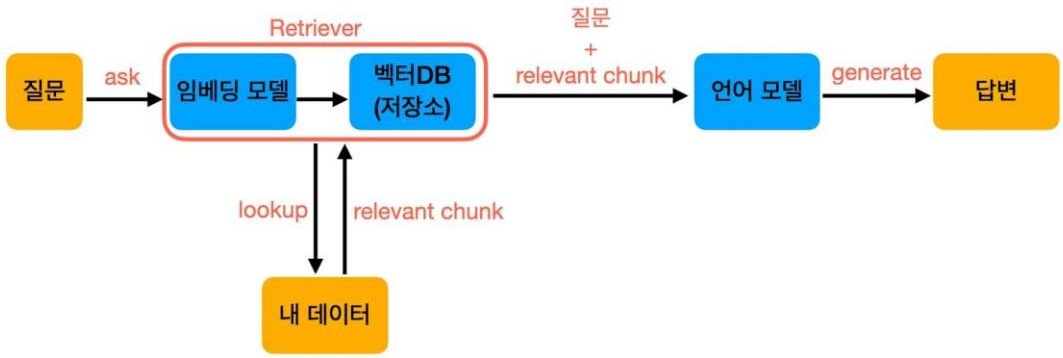


벡터스토어: RAG어플리케이션

❑ RAG 개념

RAG 구현 과정

1. **정보검색 :** 사용자의 질문에 관련된 정보를 외부 지식 소스(예: 데이터베이스나 인터넷)에서 검색
2. **질의처리(벡터화 및 유사도 검색):** 검색된 정보를 벡터 형식으로 변환하고, 유사한 정보를 찾아냄
3. **응답 생·** 텍스트를 생성

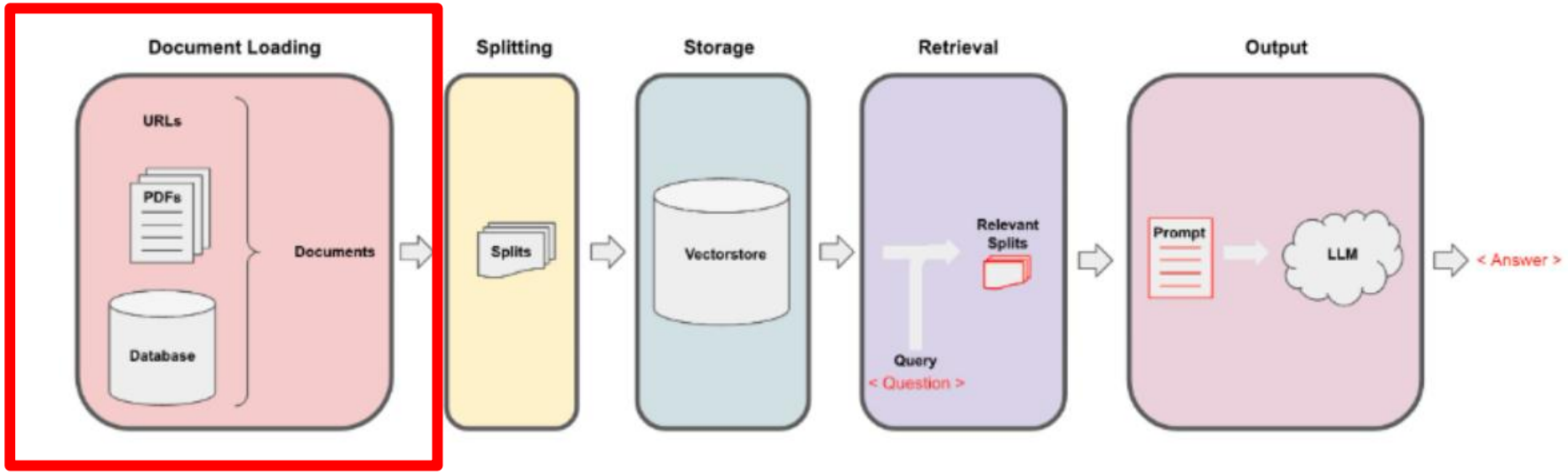


벡터스토어: RAG어플리케이션

❑ RAG 구현과정

1. 데이터 로드 및 전처리

- 데이터 업로드: PDF,Word 등 다양한 형식의 문서를 시스템에 업로드
- 전처리: 문서를 분석하기 쉬운 작은 단위(chunk)로 분할

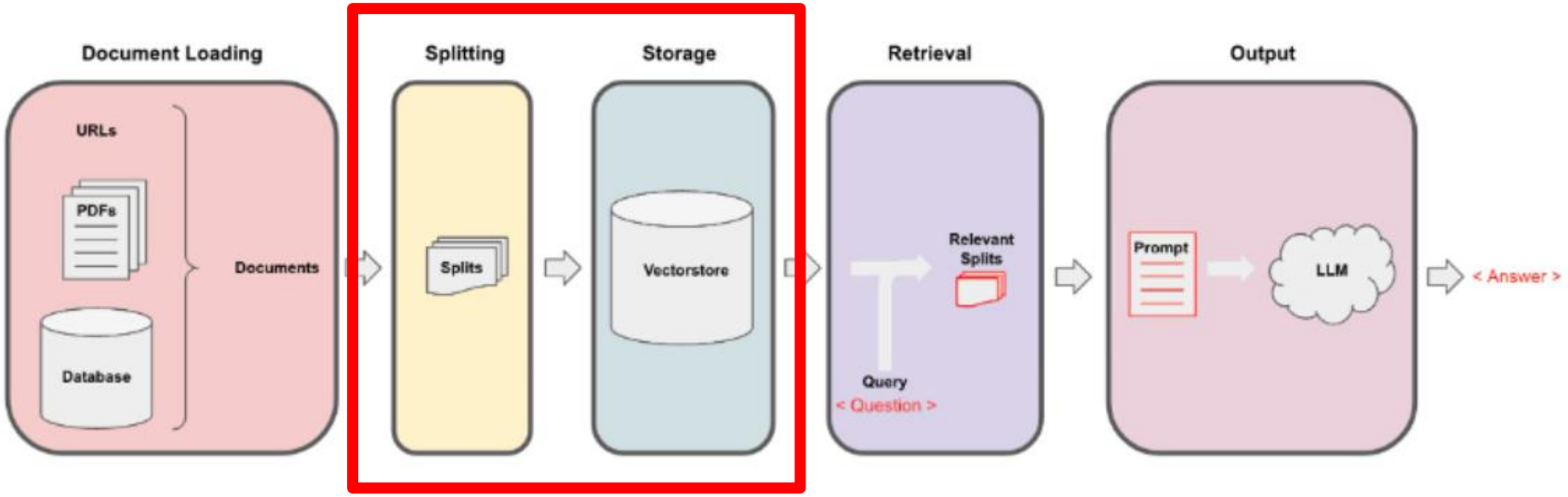


벡터스토어: RAG어플리케이션

❑ RAG 구현과정

2. 임베딩 및 벡터스토어 생성

- 임베딩: 문서의 텍스트를 숫자 벡터로 변환
- 벡터스토어 생성: 변환된 벡터를 데이터베이스에 저장

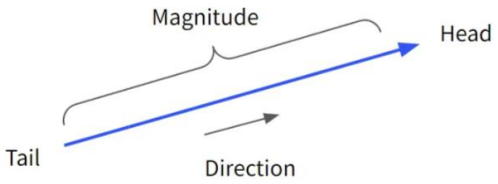


벡터스토어: RAG어플리케이션

❑ RAG 개념

벡터(Vector)

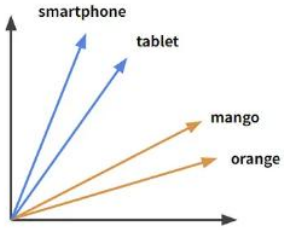
- **정의**: 길이와 방향을 가지는 수학적 객체
- **표현**: n차원 공간에서 길이가 n인 1차원 숫자 배열
- **데이터 과학**: 데이터의 특성이나 속성을 나타내는 숫



임베딩(Embedding)

- **정의**: 데이터를 모델이 이해할 수 있는 벡터 형태로 변환
- **표현**: 비정형 데이터(텍스트)를 정형 데이터인 벡터로 변환
- **의미 보존: 유사한 데이터는 벡터 공간에서 근접하게 위치**



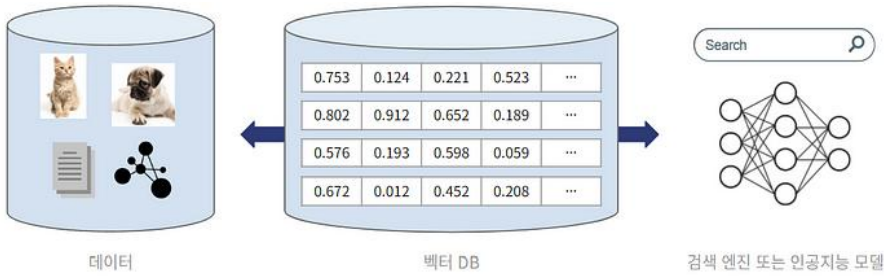


벡터스토어: RAG어플리케이션

❑ 벡터 데이터베이스

벡터 데이터베이스 (Vector DB)

- **정의**: 텍스트 데이터를 숫자벡터 형태로 저장하는 데이터베이스. 벡터 값을 데이터베이스(DB)에 저장하고 유사도에 따라 효율적으로 검색하는 역할을 하는 것이 벡터 DB임
- **일반 DB와 차이점**
  - **일반 DB:** 특정 값을 할당하여 인덱싱하고, 정확히 일치하는 결과를 반환.
  - **벡터 DB:** 임베딩을 통해 벡터로 저장하고, 유사도 기반 알고리즘을 사용하여 유사한 일치 결과를 반환.
- **특징**
  - 벡터 데이터의 양이 많아도 빠른 검색 속도 유지.
  - 생성 모델의 한계를 보완하는 검색 증간 생성(RAG)프레임워크에 필수적.
- **핵심기술**
  - 근사 최인접 이웃(ANN) 검색, 임베딩



벡터스토어: RAG어플리케이션

❑ RAG 개념

벡터 데이터베이스 (Vector DB)

- **종류**

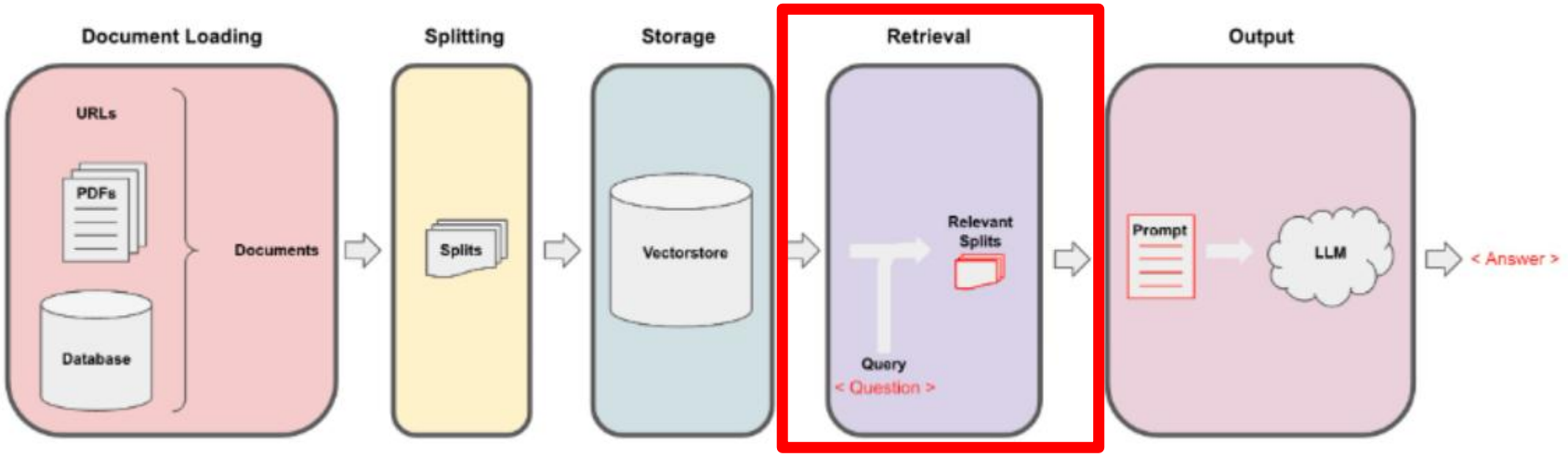
데이터베이스	유형	주요 기능	특징
Pinecone	클라우드 서비스 (유료)	중복 데이터 찾기, 순위 정하기, 데이터 검색, 분류	별도 관리 없이 사용 가능, 빠르고 정확함
Chroma	무료 오픈 소스	텍스트를 숫자로 변환, 비슷한 데이터 찾기	언어 모델 앱 개발을 쉽게 함
Weaviate	무료 오픈 소스	AI로 데이터 검색, 자동 분류	머신러닝 데이터 저장에 적합
Qdrant	무료 오픈 소스	다양한 언어 지원, 고급 필터링, 확장성	유사한 데이터 쉽게 찾기, 대용량 데이터 처리 가능
Faiss	무료 오픈 소스	빠르고 효율적인 데이터 검색	Facebook이 개발, RAM과 디스크에 저장 가능

벡터스토어: RAG어플리케이션

❑ RAG 구현과정

3. 프롬프트 및 리트리버 생성

- 프롬프트 생성: 프롬프트는 언어 모델에게 답변을 생성하도록 요청하는 문장
- 리트리버 생성: 사용자가 입력한 질문을 텍스트를 이해하고, 이를 기반으로 가장 유사한 문서를 찾기 위해 사용자의 질문도 마찬가지로 숫자 벡터로 변환
- 문서검색: 리트리버가 사용자의 질문 벡터를 데이터베이스에 있는 문서 벡터들과 비교하여, 가장 유사한 문서를 찾아줌

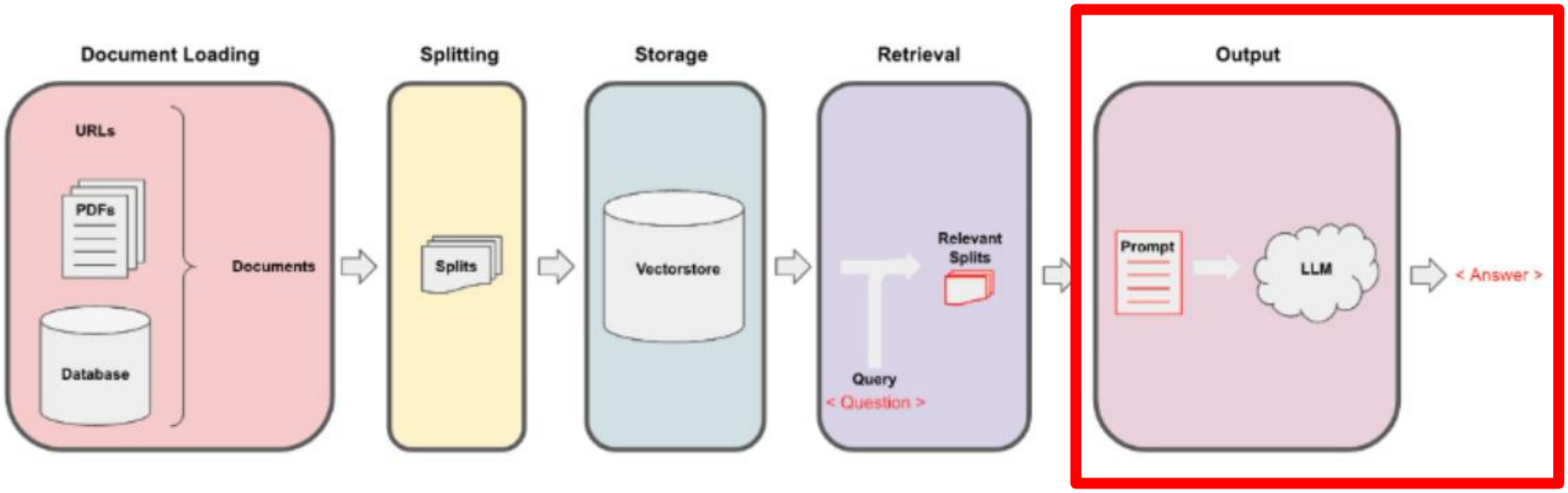


벡터스토어: RAG어플리케이션

❑ RAG 구현과정

4. 언어 모델 생성 및 체인 실행

- 생성된 프롬프트에서 리트리버가 관련 문서를 찾아 사용자 질문과 결합함. 결합된 내용에 추가 정보를 더해 언어 모델이 더 나은 답변을 생성



벡터스토어: RAG어플리케이션

❑ RAG 구현과정

4. 언어 모델 생성 및 체인 실행

- 답변 생성을 위해 기존의 언어 모델을 활용하고, 모든 구성 요소를 연결하여 워크플로우를 생성하고 실행.
- 워크플로우(체인): 정보 검색, 텍스트 생성, 벡터화 및 유사도 검색 등의 단계를 순차적으로 연결한 작업 흐름을 의미

