Contents

우리은행 비정형 데이터 자산화 시스템: 2022년 vs 2024년 임베딩 모델, 벡터 검색엔진, 벡터DB 비교 종합 보고서 1
개요 . . . 1
2022년 우리은행 시스템: 기술 구성 및 특징 . . . 1
임베딩 모델 . . . 1
벡터 검색엔진 및 벡터DB . . . 1
도입 배경 및 한국 기업 트렌드 . . . 2
2024년 우리은행 시스템: 최신 기술 구성 및 특징 . . . 2
임베딩 모델 . . . 2
벡터 검색엔진 및 벡터DB . . . 2
도입 배경 및 업그레이드 이유 . . . 3
기술 상세 비교 및 트렌드 분석 . . . 3
전체 아키텍처 관점 비교 . . . 3
업계 흐름 및 글로벌 대비 . . . 3
결론 . . . 3
Sources . . . 4

우리은행 비정형 데이터 자산화 시스템: 2022년 vs 2024년 임베딩 모델, 벡터 검색엔진, 벡터DB 비교 종합 보고서

개요

본 보고서는 우리은행의 비정형 데이터 자산화 시스템에 대해 2022년 제안서와 2024년 제안서를 바탕으로 사용된 임베딩 모델, 벡터 검색엔진, 벡터 데이터베이스(벡터DB)의 기술적 차이, 개선 배경, 각 기술의 특장점과 도입 근거를 심층적으로 비교·분석한다. 국내외의 대표적 최신 트렌드 및 국내 금융업계 실무 기술적 사례(특히 한국어에 특화된 정보)를 함께 조명하여, 실질적인 기술 변화와 전략적 방향성을 다각적으로 설명한다.

2022년 우리은행 시스템: 기술 구성 및 특징

임베딩 모델

- **Accu.TA 한국어 Transformer 언어모델**: 우리은행 자체 개발(또는 커스터마이즈)된 모델로, 한국어 금융 데이터에 특화된 언어 표현력을 강화함.
- **DPR (Dense Passage Retrieval) 구조**: Dual Encoder 방식(Passage Encoder, Question Encoder) 사용, 문서와 질문을 각각 임베딩하여 유사도 검색에 효과적.
- **DHR (Dense Hierarchical Retrieval)**: 계층적 dense 검색으로 규모가 큰 데이터셋에서 정밀한 탐색 지원.

강점 및 특징

- 도메인 데이터(금융) 전용 커스터마이즈, BM25 등 전통적 텍스트 검색 대비 정확도 대폭 개선(BM25: 54.08, DPR 비학습: 83.67, DPR 학습: 92.1, Top20 기준).
- 한국어 업무/법령 등 특수 문서에서 실제 활용 결과, 상용 키워드 기반 검색 대비 약 38%P, 도메인 특화 미적용 dense 검색 대비 8%P 정확도 향상.

벡터 검색엔진 및 벡터DB

- **FAISS**(Facebook AI Similarity Search)
  - 대규모 벡터 데이터셋의 유사도/근접도 빠른 검색을 위한 오픈소스 라이브러리.
  - GPU 지원, ANN(Approximate Nearest Neighbor) 탐색 등 강점.
  - 그러나 메타데이터 필터링 및 분산 아키텍처는 부족, File-based index 재로딩 시 성능 저하.
- 부가적으로, 인덱스DB는 기본적인 키워드 검색과 최소 메타정보 저장의 역할만 수행.

강점 및 특징

- 오픈소스 기반, 성능이 우수하며 Python 등 생태계 연계성 뛰어남.
- 빠른 프로토타이핑/파일 기반 대규모 실험 적합.

한계

- 동적 메타필터 지원 부재, 단순 ID/임베딩 저장만 지원.
- 인프라 확장/분산 아키텍처 미흡, RAG 및 하이브리드 검색 구현 어려움.

도입 배경 및 한국 기업 트렌드

- **도입 논리**: 오픈소스 위주, 한국어 금융 도메인 맞춤형 모델, 신속한 적용 및 실험이 가능한 환경 우선.
- **업계 동향**: 2022년 국내 기업들은 FAISS, Redis VSS, KoBERT 등 오픈소스 기반 임베딩·벡터 검색 실험단계였으며, 메타데이터 기반 필터, 복합검색(DB+Vector) 기능은 미성숙한 상태였다.
- **대규모 배포 변수**: Redis VSS 등은 초기에 빠른 PoC용으로 쓰였으나, 생산성/대규모 서비스에는 적합하지 않다는 평가[1][2][4].

2024년 우리은행 시스템: 최신 기술 구성 및 특징

임베딩 모델

- **M3 Embedding**
  - Dense + Sparse + Multi Vector 통합 지원(혼합/단일/복합 임베딩).
  - 100개 이상의 다국어 지원, 도메인별 파인튜닝 및 성능 최적화 가능.
- **E5 (Multilingual E5)**
  - Microsoft에서 공개한 오픈 소스, 대규모 다국어/다도메인 텍스트 페어 학습을 통해 MTEB(Massive Text Embedding Benchmark) 상위권 성능 입증.
  - Instrcut 버전 등 파생모델도 제공, 한국어·아시아권 언어 표현 우수.

강점 및 특징

- 헬스케어, 금융 등 실제 국내 사례에서 M3: 97.6% 리트리벌 정확도, Multilingual-E5-Large: 89.7% 기록(10ms 이하 응답속도)[11][12][13].
- 최신 RAG, Recency 필터, Hybrid(키워드+임베딩) 검색에 최적화, 다국어 및 도메인-커스텀 성능 강점.

벡터 검색엔진 및 벡터DB

- **Qdrant**
  - Rust 기반 오픈소스, 분산/병렬 처리를 통한 초고속/고가용 검색 서비스 제공.
  - 다양한 메타데이터(날짜, 업무구분, 카테고리, 권한, 키워드 등) 저장/동적 필터링.
  - Dense, Sparse, Multi-vector 등 다양한 형태의 벡터 동시에 저장/검색 가능.
  - Write Ahead Log, 인덱스 분할 세그먼트, 실시간 업데이트, Blob File Storage 등 대용량/실시간 서비스 아키텍처 지원.
  - API/SDK 제공, LangChain 등과의 통합으로 RAG 및 LLM 지원 수월.
  - Docker, 온프레미스, 클라우드(무료 버전 포함) 등 디플로이 옵션 다양.

이전 대비 개선점

- **메타필터 지원**: FAISS는 ID/벡터만 저장 → Qdrant는 도메인, 날짜, 권한, 키워드 등 복합 조건 검색 가능.
- **인덱싱/업데이트**: File-기반 일괄 재로딩 필요 → 실시간 인덱스/즉시 검색 지원, 부분 업데이트 및 recency filter 가능.
- **확장성/분산처리**: 단일 노드/메모리 의존 → 분산/병렬 노드 구성, 대용량 연산에도 성능 유지.
- **RAG 파이프라인**: Hybrid/Dense/Sparse/Multi-vector 통합, downstream LLM 및 고정밀 재정렬(rerank), 시맨틱 라우터 등 고도화.

도입 배경 및 업그레이드 이유

- **임베딩 모델**: 글로벌 벤치마크 및 국내 실제 테스트에서 M3, E5 계열이 기존 DPR, BERT 계열보다 한국어/다국어 정확도 및 연계성 우위 입증.
- **벡터DB 전환**: 실무에서 메타데이터 조건별 검색, 실시간/대규모 서비스, RAG 통합, 운영/유지관리 편의성 등 실질적 수요가 확대 → Qdrant와 같은 차세대 솔루션 필요성 부각.
- **재정렬/필터 등 부가 기능**: 최신 LLM/RAG 파이프라인 적용을 위한 재정렬, 신선도 기반 데이터 우선 정렬 등 최신 검색 시나리오 적극 반영.

기술 상세 비교 및 트렌드 분석

전체 아키텍처 관점 비교

항목	2022년 시스템	2024년 시스템
임베딩 모델	Accu.TA 한국어 Transformer, DPR, DHR	M3, Multilingual E5, BGE-M3 등 최신 다국어/다기능 모델
검색엔진/DB	FAISS + (단순 인덱스DB, File기반)	Qdrant(분산, 동적 메타필터, Hybrid), IndexDB(키워드, 메타 보조)
정확도	Top20 기준: BM25(54.08), DPR(92.1)	MIRACL nDCG@10: mE5-large(66.5), 실제 업무 2차 구축에서 8%P↑
메타데이터	지원 미흡(최소 ID 기반)	다양한 속성/권한/시간 등 필드별 동적 조건검색 지원
확장성	단일 노드/메모리 규모 한계	분산, 병렬, 실시간 업데이트, 대규모(수십억 벡터) 저장/검색
통합/연계	단순 검색/답변 생성	LLM/RAG, 비즈니스 보고서, 챗봇, 내부 규정/검색/상담 등 서비스 연계
배포방식	오픈소스(Python 위주, 단일 서버)	오픈소스+클라우드+온프레미스, API/SDK 다양화, 실시간 데이터 연동

업계 흐름 및 글로벌 대비

- 국내 은행, 통신, 대형 IT 기업 등도 2022년까지는 File-based 검색엔진(FAISS, Redis VSS), KoBERT 등 오픈소스 임베딩 활용이 주였으며, 2023~2024년 LLM/RAG 대중화와 함께 Qdrant, Milvus, Pinecone, Weaviate 등 프로덕션급 벡터DB로 전환하는 추세[6][7][8].
- 글로벌 트렌드는 ANN 검색 성능(지연 10~30ms), 대규모벡터 저장·검색, 분산·하이브리드 검색·메타데이터 확장성 중심으로 진화.

결론

2022년 우리은행 비정형 데이터 자산화 시스템은 오픈소스 기반 한국어 특화 임베딩(DPR, Accu.TA)과 FAISS 등 전통적 벡터 검색엔진을 사용, 단순한 메타데이터로 높은 검색 정확도와 빠른 실험을 지향했다. 한계는 메타필터링, 실시간/분산 운영, RAG/LLM 통합 부재에 있었다.

2024년 시스템은 M3/E5와 같은 세계 수준의 멀티링구얼·다기능 임베딩 모델, 그리고 Qdrant 벡터DB를 도입하며, 복합 메타데이터 기반 고정밀 검색, 대규모 분산 처리, 실시간 인덱싱, LLM/RAG 파이프라인 연동 등 현대적 기능을 갖춘 대규모 AI 데이터 자산화 인프라로 도약했다.

이는 오픈소스 생태계의 혁신, 한국어 정보처리 고도화, 대규모 서비스 최적화 요구에 따라 자연스럽게 진화한 결과로 평가된다.

Sources

[1] 2024-04-22_우리은행_비정형 데이터 자산화 시스템 2단계 구축_Ⅳ.기술부문.pdf
[2] 2024-04-22_우리은행_비정형 데이터 자산화 시스템 2단계 구축_Ⅳ.기술부문.pdf
[3] 2024-04-22_우리은행_비정형 데이터 자산화 시스템 2단계 구축_Ⅳ.기술부문.pdf
[4] 2022-11-21_(주)우리은행_우리은행 비정형 데이터 자산화 시스템 구축_Ⅳ. 기술 부문.pdf
[5] 2022-11-21_(주)우리은행_우리은행 비정형 데이터 자산화 시스템 구축_Ⅳ. 기술 부문.pdf
[6] Qdrant 공식 문서: https://qdrant.tech/documentation/
[7] Qdrant 기술 소개: https://qdrant.tech/
[8] Qdrant Fundamentals: https://qdrant.tech/documentation/faq/qdrant-fundamentals/
[9] LangChain - Qdrant 통합: https://docs.langchain.com/oss/python/integrations/vectorstores/qdrant
[10] Langflow - Qdrant: https://docs.langflow.org/bundles-qdrant
[11] M3 임베딩 논문: https://www.researchgate.net/publication/384220744_M3-Embedding_Multi-Linguality_Multi-Functionality_Multi-Granularity_Text_Embeddings_Through_Self-Knowledge_Distillation
[12] 도메인 특화 임베딩 비교 연구: https://arxiv.org/html/2502.07131v1
[13] 의료 RAG 적용 연구: https://pmc.ncbi.nlm.nih.gov/articles/PMC12370418/
[14] Massive Multilingual Text Embedding Benchmark: https://huggingface.co/papers?q=Massive%20Multilingual%20Text%20Emb
[15] 국내 RAG 임상 평가: https://synapse.koreamed.org/articles/1516092141