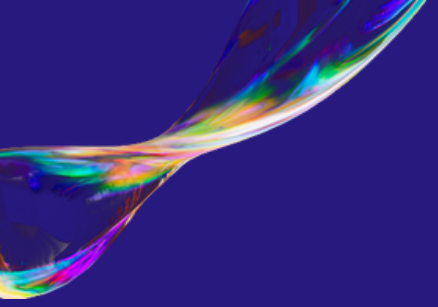
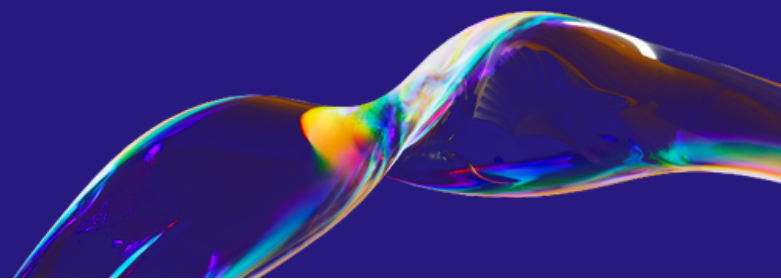


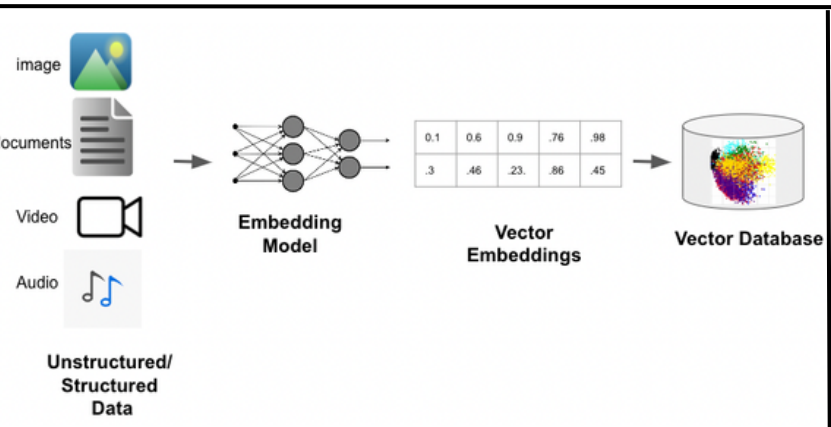
GENERATIVE AI:

VECTOR DATABASES & RETRIEVAL AUGMENTED GENERATION

ŞEVKET AY



Vector Databases



Gömüler, genellikle yapılandırılmamış verilerin sayısal formatta temsil edildiği n boyutlu bir veri evreninde bulunur. Ancak geleneksel veri tabanları bu vektör gömülerini depolamak için uygun değildir. İşte burada devreye Vector Store veya Vector Databases girer. Bu özel veri tabanları, vektör gömülerini etkili bir şekilde depolamak ve almak için optimize edilmiştir, farklı modellere ve arama algoritmalarına göre çeşitlilik gösterirler.

Popüler Vector Databases & Libraries

ChromaDB

- Open Source ve On-Prem
- Embedding için Chroma all-MiniLM-L6-v2 kullanır fakat başka bir embedding modelde kullanılabilir.
- Esnek sorgulama
- Ölçeklenebilirlik ve performans açısından sınırlı
- Yaklaşık En Yakın Komşu aramasına dayanır.

Pinecone

- Cloud ve kullanım başına ödeme
- Hızlı arama özelliği
- Filtrelenmiş vektör araması
- Dinamik indeksleme
- Veri gizliliği problem olabilir.
- Yaklaşık En Yakın Komşu aramasına dayanır.

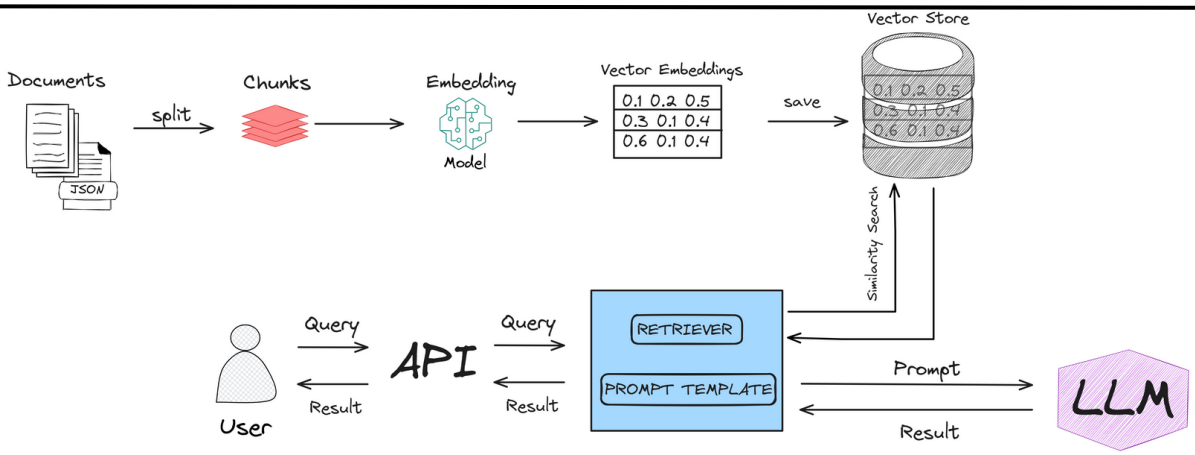
Milvus

- Open Source ve On-Prem
- Tek başına bir vektör veri tabanı değildir.
- Ölçeklenebilirlik odaklı
- Kolay entegrasyon
- Hızlı arama özelliği
- L2 Öklid mesafesini hesaplar.

Vector DB Seçerken Dikkat Edilmesi Gerekenler

Ölçeklenebilirlik: Veri tabanı büyük hacimli yüksek boyutlu verileri işleyebilir ve veri ihtiyaçlarınız büyüdükçe buna adapte olabilir mi?
Performans: Veri alımı, arama işlemleri ve vektörler üzerinde sorunsuz bir performans sunabilen hızlı bir veri tabanı seçilebilir.
Esneklik: Farklı veri türleri ve formatlarıyla uyumlu, çeşitli kullanım senaryolarına adapte olabilen bir veri tabanı seçimi yapılabilir.
Kullanım Kolaylığı: Karmaşık süreçlerden kaçınmak adına kullanıcı dostu kurulum, sezgisel API'ler ve detaylı belgelerle desteklenmiş bir veri tabanı tercih edilebilir.
Güvenilirlik: Bulut tabanlı depolama çözümleri ölçeklenebilirlik ve yedeklenebilirlik sunarken, veri gizliliği ve uyumluluk konularında endişelere neden olabilir.

Retrieval Augmented Generation



Niçin RAG?

İhtiyacımıza yönelik bir LLM kullanmamız gerektiğinde ilk aklımıza gelen dil modeline ince ayar yapılması olabilir fakat LLM'i eğitmek için gerekli olan hesaplama altyapısı, yüksek kaliteli ve alana özgü veri kümelerinin bir araya getirilmesi, halüsinasyon problemi ve bağlam penceresi gibi unsurlar modele ince ayar yapılmasını zorlaştırabilir. Tam da bu noktada RAG devreye girebilir.

RAG, harici bir veri tabanından güncel veya bağlama özel verilerin getirilmesi ve bir LLM'den bir yanıt oluşturmasını isteyerek bu sorunu çözmesi istendiğinde bu verilerin kullanılabilir hale getirilmesi anlamına gelir.

Faydaları

Halüsinasyonların Azaltılması: RAG, bağlı olduğu dış kaynaklar içerisinden bilgileri alması ile halüsinasyonu azaltabilir.
Güncel Bilgiler: Dış kaynaktaki veriler, önemli maliyetlere maruz kalmadan sürekli olarak güncellenebilir.
Uyarlanabilirlik: RAG, kuruluşlar içindeki özel veri tabanları ile sorunsuz bir şekilde bütünleşebilir.
Denetlenebilirlik: RAG'ın vektör veri tabanındaki bilgilerin kaynağı belirlenebilir. Veri kaynakları bilindiği için RAG'daki yanlış bilgiler düzeltilebilir veya silinebilir.

Sınırlamaları

Dış Kaynaklara Bağımlılık: Dış kaynaklar bilgi zenginliğini artırabilirken aynı zamanda çıktıların kalitesi ve doğruluğu bu kaynaklara bağlıdır.
Aşırı Bilgi Yüklemesi: Alınan bilgileri modelin kendi bilgisiyle dengelemek zor olabilir ve bu durum ayrıntılı veya daha az tutarlı yanıtlara yol açabilir.
Önyargı ve Yanlış Bilgi: Kullanılan dış kaynaklara bağlı olarak önyargıları sürdürme veya yanlış bilgi yayma riski vardır.
Gecikme Sorunları: Harici veri tabanlarından bilgi alma süreci, gerçek zamanlı uygulamalarda çok önemli olan LLM'lerin yanıt verme hızını etkileyen gecikmeye neden olabilir.

Daha iyi yanıtlar alabilmek için retrieval performansını nasıl geliştirirsiniz?

1) Embedding modelini akıllıca seçmek. Seçim yaparken sorulabilecek bir soru: "Embedding modeli elinizdekine benzer veriler üzerinde eğitilmiş mi?"

2) Veri tabanınızın embedding alanı, kullanıcı sorgularını içeren tüm verileri kapsamalıdır. Örneğin, filmlerle ilgili bir veri kümeniz varsa ve tıbbi bir soru sorarsanız, arama sisteminin performansı olumsuz etkilenebilir. Bu nedenle, vektör veritabanındaki belgelerin sorgularınızla uyumlu olduğundan emin olun.