Reranker Raporu

1) Amaç

Sorguya göre belgeleri önem sırasına koymak.

- Akış: **Belgeleri al → API'ye gönder → skorları al → sırala → top_n uygula.**
- Hata yönetimi: API başarısız olursa orijinal belgeler korunur.
- Uyarlamalar: Cohere, Infinity AI veya Azure üzerinden çalışabilir; mantık aynı, sadece client farklıdır.

2) Embedding vs Reranker

Özellik	Embedding (Bi-Encoder)	Reranker (CrossEncoder)
Girdi	Sorgu ve doküman ayrı encode	Sorgu + doküman birlikte encode
Hız	Çok hızlı (ANN/FAISS/Chroma)	Daha yavaş (her çift için forward)
İsabet	Orta–yüksek (kaba sıralama)	Yüksek (ince sıralama)
Kullanım	İlk candidate retrieval	Sonraki re-ranking aşaması
Maliyet	Düşük	Daha yüksek (çift sayısına bağlı)

Sonuç: Embedding hızlı ama kaba filtreleme için; Reranker ise daha pahalı ama çok daha doğru sonuçlar için son katman.

3) Base Reranker Yapısı

- **Soyut sınıf**: `rerank` fonksiyonunu tanımlar, alt sınıflarda override gerekir.
- **Pydantic BaseModel'den türetilmiş** → veri doğrulama, ayar yönetimi kolay.
- **Özellikler:**
  - `arbitrary_types_allowed=True` → Python tipleri desteklenir.
  - `populate_by_name=True` → attribute'lar isimle doldurulur.
- **rerank fonksiyonu:** Abstract → CohereReranker, InfinityReranker, SentenceTransformerReranker gibi somut sınıflar uygular.

4) Cohere Reranker

Kullanım Senaryosu (NLQ → SQL + Semantic):

- SQL'den gelen sonuçlar (örn. `age > 25 AND skill="Python"`).
- Semantic search'ten gelen embedding sonuçları.
- Problem: İki liste birleştiğinde en alakalı adayın kim olduğunu anlamak zor.
- Çözüm: Tüm aday profilleri `Document(content=...)` olarak Cohere'a gönderilir → relevance score'a göre sıralanır → hibrit kalite artışı.

Özellikler:

- Multilingual destek: `rerank-multilingual-v3.0` (100+ dil, Türkçe dahil).
- Yeni modeller: `rerank-v3.5` (4096 token).
- Skor: [0–1] normalize.
- Mantık: Query + doküman chunk'lara bölünür, her chunk için skor hesaplanır, en yüksek chunk skoru dokümanın final skorudur.

Entegrasyon:

- Girdi: query + belge listesi (string veya YAML).
- Çıktı: relevance score eklenmiş belgeler.
- Hata olursa: belgeler değişmeden döner.

Zorluklar:

- Uzun doküman → chunking gerekir.
- Cohere Python client sürüm uyumluluğu → güncel sürüm kurulmalı.
- API key güvenliği → environment variable kullanılmalı.
- Performans: Çok fazla aday gönderilirse maliyetli → önce Top-K filtre, sonra rerank.

5) Infinity Reranker

- HuggingFace modellerini (örn. `BAAI/bge-reranker`) **kendi sunucunda** çalıştırmanı sağlar.
- **Avantajları:**
    - Veri gizliliği (dışa çıkmaz).

    - API maliyetinden bağımsız.
    - GPU/CPU üzerinde düşük gecikmeli.
- **Mantık:** Cohere ile aynı akış, sadece client Infinity AI. Senkron/asenkron seçenekleri vardır.
- **Kullanım Alanları:**
    - Yüksek gizlilik gereken kurum içi uygulamalar.
    - API maliyetini azaltmak.
    - Offline GPU/CPU çalıştırma.

6) SentenceTransformer Reranker

- `sentence-transformers` kütüphanesi ile CrossEncoder tabanlı.
- Her sorgu–doküman çifti için skor üretir, Top-N seçilir.
- Metadata'ya `reranking_score` eklenir → analiz kolaylaşır.
- **Faydaları:**
    - Doğruluk artışı: CrossEncoder daha iyi bağlam anlar.
    - Hızlı filtreleme: büyük veri kümelerinde top_n ile performans kazanımı.
    - Yeniden kullanılabilirlik: Farklı pipeline'lara kolay entegre olur.
    - Analitik: Skorlar log'lanır, izleme kolaydır.
    - Hata toleransı: Model çökse bile belgeler orijinal haliyle döner.

Riskler:

- Büyük veri setlerinde inference süresi uzun → batching şart.
- Top_n yanlış ayarlanırsa (çok küçük / çok büyük) kalite kaybı olur.
- Modeller güncellenmeli, periyodik benchmark yapılmalı.

7) Agno'da Strateji

Neden iki reranker öncelikli?

- Cohere Reranker: SaaS API, kolay entegrasyon, yüksek doğruluk.
- SentenceTransformers: Açık kaynak, offline, maliyetsiz, GPU ile hızlı.

→ Böylece hem **enterprise production** (Cohere) hem **akademik/prototip/offline** (ST) kullanım kapsanıyor.

Gold Standard:

- Literatürde en çok benchmark edilenler Cohere Rerank & HuggingFace CrossEncoders.
- BM25, GPT promptlama, VoyageAI → daha niş, henüz oturmamış.
- Agno: önce temel (stabil, yaygın, kolay setup) modelleri seçmiş.

8) Ölçüm ve Kalibrasyon

- **Offline değerlendirme:** NDCG@K, MRR@K, Recall@K, HitRate@K
- **Kalibrasyon:** 30–50 tipik sorguda skor dağılımına bak → uygun eşik seç.
- **Online A/B test:** küçük kitle üzerinde tıklama / işlem metriği.

9) Sonuç

- Embedding hızlı kaba sıralama için, Reranker ise kesin kalite için kullanılır.
- Cohere: kolay, güçlü ama maliyetli ve dışa bağımlı.
- Infinity/ST: offline, esnek ama optimizasyon sizde.
- Agno'nun yaklaşımı → her iki dünyanın (cloud API + open source) avantajlarını aynı sistemde toplamak.