Reranker Teknik Raporu

1. Reranker Nedir?

- **Amaç:** Arama motorunun getirdiği ilk 50–200 sonucu yeniden sıralamak, en alakalıları en üste taşımak.
- **Nasıl:** Sorgu (query) ve her aday doküman birlikte modele verilir → model tek bir skor üretir.
- **Skor Aralığı:**
    - Sigmoid: 0–1
    - Identity: Serbest aralık (ör. -10…+10)
    - Tanh: -1…1 (scale ile genişletilebilir)
- **Neden:** Embedding hızlıdır ama kaba sonuç getirir. Reranker ise ince eleyip sık dokur ve doğru sıralamayı sağlar.

2. Neden CrossEncoder?

- **Bi-encoder (embedding):** Query ve doküman ayrı encode edilir, cosine similarity ile eşleşme yapılır → hızlı ama bağlam ilişkileri zayıf.
- **CrossEncoder (reranker):** Query + doküman aynı transformer içine verilir, attention çapraz kurulur → bağlamlar arası etkileşim çok daha güçlü.
- **Dezavantaj:** Her aday için ayrı çalışır (ör. k=100 doc → 100 forward pass).
- **Pratik Kullanım:**
    - Bi-encoder ile top-K (örn. 100) sonuç getir.
    - CrossEncoder ile bu K dokümanı rerank et.
    - Kullanıcıya en iyi 5–10 sonucu sun.

3. Eğitim (Loss Fonksiyonları)

- **Binary CrossEntropyLoss (BCE):**
    - Her query–doc çifti için 1 (alakalı) / 0 (alakasız).
    - Basit, güçlü, eğer elinde etiketli pozitif/negatif varsa ilk tercih.
- **Cached Multiple Negatives Ranking Loss (CMNRL):**

    - Pozitif çiftler var ama negatif etiket yoksa, batch içindeki diğerleri otomatik “yanlış” sayılır.
    - In-batch negatives + GradCache sayesinde verimli.
    - Büyük veri setlerinde çok faydalı.
- **LambdaLoss (listwise):**
    - Sıralama bilgisini (aynı query için doc1 > doc2 gibi) optimize eder.
    - Eğer elinde listwise etiket varsa en iyi yöntem.

Seçim:

- Açık pozitif/negatif → **BCE**
- Sadece pozitif, çok data → **CMNRL**
- Sıralı liste → **LambdaLoss**

4. Skorların Yorumu

- **Sigmoid + scale=10:** Skorları keskinleştirir.
- **Identity + scale=1:** Ham logit değerleri.
- **Tanh + scale=10:** mGTE gibi modellerde önerilmiş.

Mutlak skor yerine **göreli sıralama** güvenilirdir.

5. Inference (Kullanım)

```
from sentence_transformers import CrossEncoder

model = CrossEncoder("tomaarsen/reranker-ModernBERT-base-gooaq-bce")

pairs = [[query, doc1], [query, doc2], ...]
scores = model.predict(pairs)

# Kolay mod
model.rank(query, [doc1, doc2, ...])
```

- Her (query, doc) için tek skor döner.
- Skor yüksekse doküman daha alakalıdır.
- **Top-K seçimi:** 50–100 iyi denge. 200+ kalite artar ama latency yükselir.

- **Uzun metin:** Token limitine dikkat; gerekirse dokümanları passage'lara böl.
- **Dil desteği:**
    - İngilizce: ModernBERT, MS MARCO tabanlı modeller.
    - Çok dilli: multilingual cross-encoder veya Cohere Rerank.

6. Veri Hazırlama

- **BCE için çift:** query, answer, label (0/1)
- **CMNRL için triplet:** query, positive, negative
- **LambdaLoss için liste:** query, [doc1, doc2, ...] + sıralama bilgisi

Negatif üretme (pratik):

- Embedding ile top-20 doküman getir.
- Uzman alakalı / alakasız işaretlesin.
- Geri kalanları otomatik negatif sayabilirsin.

Dataset boyutu:

- İlk deneme: 1K–5K çift yeterli.
- Büyük veri (50K–100K) → daha stabil performans.

7. Değerlendirme

- **nDCG@10:** En üst 10 sonucun kalite ölçümü.
- **MRR@10:** İlk doğru sonucun sırası.
- Özellikle "sınırda alakalı" örneklerle test yapmak kritik.

8. Sık Yapılan Hatalar

- **Top-K çok büyük seçmek:** Latency patlar.
- **Domain farkı:** MS MARCO gibi datasetler geneldir, senin domaininde düşüş olabilir. Fine-tuning şart.
- **Uzun doküman:** Truncation kritik bilgiyi kesebilir → passage-level rerank daha iyi.

- **Mutlak skor yorumu:** Skorlar modelden modele değişir; sıralama güvenilir olandır.

9. HR Candidate Search Senaryosu

- **Query:** "güçlü iletişim becerisine sahip adaylar"
- **Embedding/SQL:** 100 CV getirir.
- **Reranker:** Query + her CV için skor üretir.
- **Sonuç:** Gerçekten iletişim becerisi vurgulanan CV'ler üstte çıkar, yanlış pozitifler elenir.

10. Pipeline Önerisi

1. **Embedding (Azure OpenAI text-embedding-3-large)** ile top-100 getir.
2. **Reranker:**
    a. Cloud tabanlı çok dilli istersen Cohere Rerank.
    b. Offline/GPU istersen SentenceTransformers CrossEncoder (ModernBERT veya multilingual).
3. **K Denemeleri:** 50 / 100 / 150 → latency ve kalite karşılaştır.
4. **Değerlendirme:** 30–50 query için ndcg@10, mrr@10 ölç.

11. Özet

- **Embedding:** hızlı kaba filtre.
- **Reranker:** kaliteli sıralama, ince eleyip sık dokur.
- **CrossEncoder:** bu işin standardı, çünkü query ve doc'u aynı anda işler.
- **Loss fonksiyonları:** verinin etiketlenme şekline göre seçilir.
- **Üretimde pratik yaklaşım:** Embedding → top-K → CrossEncoder Reranker.