DLN Finder: Arama Motoru Uygulaması Raporu

Eyüp Dalan - 24501037

1. Giriş

Bilgiye hızlı, etkili ve doğru bir şekilde ulaşmak günümüz dijital dünyasında en kritik ihtiyaçlardan biridir. Özellikle haber, akademik içerik veya ticari bilgiye erişim açısından arama motorlarının başarısı, kullanılan sıralama algoritmalarının kalitesiyle doğrudan ilişkilidir. Bu projede, geleneksel içerik tabanlı sıralama yaklaşımlarına (örneğin BM25) ek olarak bağlantı yapısına dayalı algoritmaların (PageRank ve HITS) katkısını araştırmayı ve bunları hibrit bir modelle birleştirerek nasıl çalıştığı ve birbirine etkileri gözlemlenmeye çalışılmıştır.

Bu projede, WARC formatındaki CommonCrawl verilerinden yola çıkarak hibrit sıralama sistemine sahip bir arama motoru geliştirilmiştir. BM25, PageRank ve HITS algoritmalarının birleştirilmesiyle oluşturulan bu sistem, daha etkili bir bilgi getirme deneyimi sunmayı amaçlamaktadır. Hem içerik temelli hem de bağlantı yapısına dayalı çıkarımlarla çalışan bu hibrit model, arama sonuçlarının kalitesini önemli ölçüde artırmaktadır.

2. Veri Seti

- Veri Kaynağı: Common Crawl (2025 Ocak ayı)
  - “https://data.commoncrawl.org/crawl-data/CC-NEWS/2025/01/warc.paths.gz”
  - “https://data.commoncrawl.org/” path’i sonuna bu linkte’ki path’ler tek tek eklenerek dosyalar indirilmiştir.
- Boyut: Yaklaşık 100 GB (100 dosya)
- Filtreleme Yöntemi: Basın İlan Kurumu tarafından yayınlanan Nisan ayına ait "reklam alabilir haber siteleri" listesi kullanıldı. Bu listeden yaklaşık 1200 domain alınarak, veri filtreleme için kullanıldı.
- Sonuç: WARC dosyalarının parse edilmesi ve bu domain filtresine göre ayıklanması sonucunda yaklaşık 100.000 adet HTML sayfası elde edilmiş ve PostgreSQL veri tabanına kaydedilmiştir.
- İlgili Kod: download.py

3. Veri Önişleme

Veri önişleme aşamasında aşağıdaki adımlar uygulanmıştır:

- HTML sayfalarından saf metin ve hyperlink'ler çıkarılmıştır.
- HTML etiketleri, script ve style blokları temizlenmiştir.
- Metinler küçük harfe dönüştürülmüş, noktalama işaretleri kaldırılmış ve tek karakterli ya da alfanumerik olmayan kelimeler filtrelenmiştir.
- Stopword temizliği uygulanmış ve elde edilen metin token'lara ayrılmıştır.
- Temizlenmiş metinler ve hyperlink'ler ayrı bir PostgreSQL tablosunda saklanmıştır.
- İlgili Kod: preprocessing.py

4. BM25 Algoritması

- Tokenize edilmiş belgelerden ters indeks (inverted index) oluşturulmuş, her kelime için belge frekansları kaydedilmiştir.
- Ayrıca her doküman için toplam uzunluk bilgisi (kelime sayısı) hesaplanmış ve doc_lengths tablosuna yazılmıştır.
- BM25 formülü uygulanarak her sorgu için dokümanlara ait skorlar hesaplanmıştır.

- Bu işlemler PostgreSQL'den veri çekilerek Python tarafında gerçekleştirilmiştir. Skorlar anlık olarak sorgulara göre hesaplanmaktadır.
- İlgili Kod: inverted_index.py, bm25_implementation.py

5. PageRank

- Sayfalar arası hyperlink'lerden yönlendirilmiş bir graph modeli kurulmuştur.
- NetworkX kütüphanesi kullanılarak klasik PageRank algoritması iteratif olarak uygulanmış ve her sayfaya ait otorite puanı elde edilmiştir.
- Elde edilen skorlar pagerank adlı tabloya kaydedilmiştir.
- İlgili Kod: pagerank.py

6. HITS

- Aynı graph modeli kullanılarak HITS algoritması uygulanmıştır.
- Bu algoritma ile sayfaların hem "hub" hem de "authority" skorları hesaplanmıştır.
- Hibrit sistemde authority skorları tercih edilmiş, "hits" tablosuna kayıt yapılmıştır.
- İlgili Kod: hits.py

7. REST API

- Flask framework'ü kullanılarak bir RESTful API geliştirilmiştir.
- Kullanıcıdan gelen query, alpha (BM25), beta (PageRank), gamma (HITS) parametrelerine göre hibrit skor hesaplanmaktadır.
- Sistem, normalize edilmiş skorları ağırlıklı ortalama ile birleştirerek sonuç döndürmektedir.
- İlgili Kod: rest_api.py

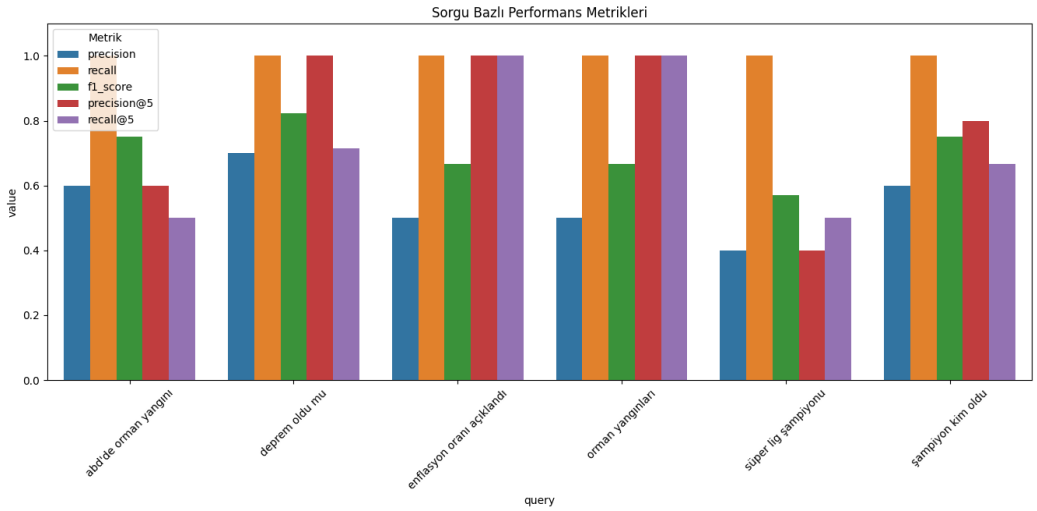
8. Web Arayüz (Next.js)

- Next.js frameworkü kullanılarak modern bir arama arayüzü geliştirilmiştir.
- Arama kutusu, ağırlık ayarları (slider'lar veya input alanları) ve arama sonuçlarının listelenmesi gibi bileşenler içermektedir.
- Kullanıcıdan gelen arama sorgusu ve seçilen ağırlıklar API'ye iletilmekte, gelen sonuçlar başlık, bağlantı ve skor bilgisiyle sunulmaktadır.
- İlgili Kod: "ui" klasörü içerisinde

9. Performans Metrikleri

- 6 farklı sorgu için ilk 10 arama sonucunun elle etiketlenmesiyle 60 belgeye ait relevance etiketi oluşturulmuştur.
- Sorular:
  - orman yangınları
  - abd'de orman yangını
  - enflasyon oranı açıklandı
  - deprem oldu mu
  - şampiyon kim oldu
  - süper lig şampiyonu
- Buna göre aşağıdaki gibi ortalama sonuçlar elde edilmiştir:

Metrik	Değer
Precision	0.55
Recall	1.00
F1-score	0.70
Precision@5	0.80
Recall@5	0.73



- İlgili Kod: evaluation_metrics.py

10. Gözlemler ve Değerlendirme

- Sistem ilgili dokümanları oldukça iyi bulabilmekte, ancak gereksiz ya da kötü sıralanıp öne çıkan belgeler precision'ı düşürmektedir.
- Bazı HTML sayfalarının "görünmeyen" elementlerinde (meta, footer, js script blokları, koda gömülmüş metin içeren gizli elementler) çöp metinler bulunması sistemin skorlamasını yanıltabilmektedir.
- HITS ve PageRank gibi yapısal sinyallerin hibrit modele dahil edilmesi, salt BM25 skorlamasından daha tutarlı sonuçlar sağlamıştır.
- Kullanıcı ağırlıkları ile skor bileşenlerini özelleştirerek farklı stratejiler denenebilmektedir.

11. Sonuç

Bu projede; veri toplama, işleme, sıralama algoritmaları, API geliştirme ve web arayüz entegrasyonu gibi farklı katmanlar bir araya getirilmiştir. BM25 gibi içerik tabanlı sıralama yöntemlerinin PageRank ve HITS gibi link tabanlı metriklerle birleştirilmesi, arama kalitesini belirgin bir şekilde artırmaktadır.