DLN Finder: Arama Motoru Uygulaması Raporu

Eyüp Dalan - 24501037

1. Proje Motivasyonu ve Amacı

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 proje, geleneksel içerik tabanlı sıralama yaklaşımlarına (örneğin BM25) ek olarak bağlantı yapısına dayalı algoritmaların (PageRank ve HITS) katkısını araştırmayı ve bunları hibrit bir modelle birleştirerek daha başarılı bir arama motoru deneyimi sunmayı hedeflemiştir.

Arama motorlarının sadece metin benzerliğine göre değil, aynı zamanda sayfaların ne kadar otoriter olduğunu da dikkate alması gerektiği varsayımı bu projenin temel motivasyonunu oluşturmuştur. Bu kapsamda, içerik ve link tabanlı algoritmaları bir arada kullanan bir hibrit sistemin performansını gözlemlemek amaçlanmıştır.

2. Kullanılan Veri Setinin Tanımı ve Kaynağı

- 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. Sistemin Genel Mimarisi

Sistem, veri toplama, önişleme, sıralama algoritmalarının uygulanması, REST API geliştirme ve kullanıcı arayüzü katmanlarını içermektedir. Aşamalar aşağıdaki gibidir:

- Veri toplama ve filtreleme (download.py)
- HTML'den içerik ve link çıkarımı, metin temizleme (preprocessing.py)
- BM25 için ters indeks ve skor hesaplama (inverted_index.py, bm25_implementation.py)
- PageRank ve HITS ile sayfa otorite skorlarının çıkarımı (pagerank.py, hits.py)
- Verilerin tutulması için db: PostgreSql
- Flask ile RESTful API sunumu (rest_api.py)
- Next.js tabanlı kullanıcı arayüzü (React - TypeScript)

4. Kullanım Senaryosu

Kullanıcı, web arayüzünden arama sorgusu girer ve BM25, PageRank, HITS için ağırlıkları belirler. Bu istek REST API üzerinden backend'e iletilir. API, bu ağırlıklara göre normalize edilmiş skorları birleştirerek ilgili belgeleri sıralar ve yanıt olarak döner. Kullanıcı, sonuçları başlık ve bağlantı halinde arayüzde görüntüler.

5. Algoritmaların Uygulanışı ve Teknik Detaylar

BM25

- 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.

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.

HITS

- Aynı grafik yapısı HITS algoritmasına uygulanmıştır.
- Authority skorları hibrit sıralama için kullanılmıştır.

REST API ve Arayüz

- 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.

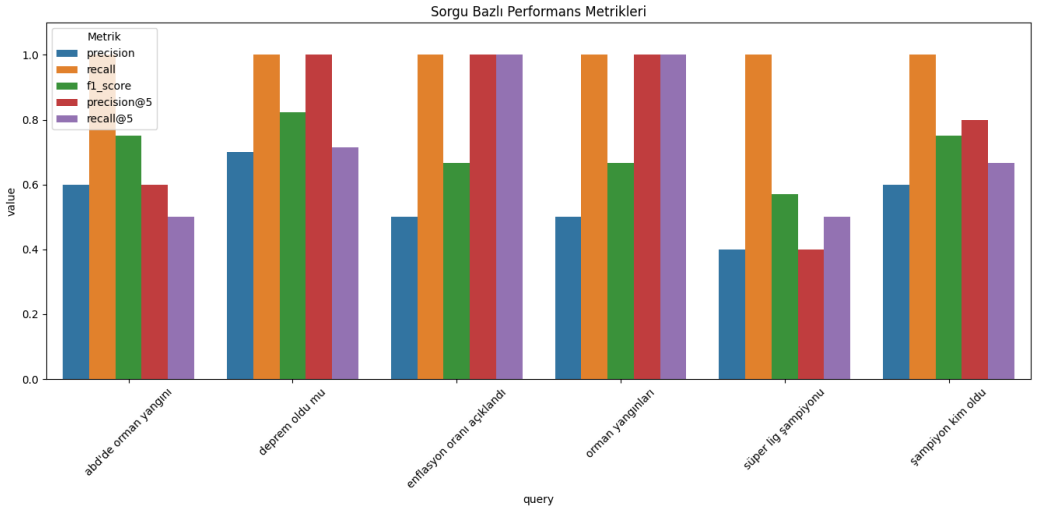
6. Karşılaşılan Teknik Zorluklar ve Çözümleri

- Veri büyüklüğü: 100GB'lık veri seti yerel ortamda işlenemezdi. Bu nedenle alt örnekleme ve domain bazlı filtreleme ile veri azaltıldı.
- Dil filtresi: Yalnızca Türkçe sayfalarla çalışmak için Basın İlan Kurumu'nun domain listesiyle filtreleme yapıldı.
- HTML karmaşıklığı: Sayfaların içeriğinde "görünmeyen" metinlerin indekslenmesi (footer'da her sayfada bulunan linkler ve metinler, gizli elementler içerisindeki metinler, SEO çalışmaları için eklenmiş gereksiz metinler, otomatik üretilmiş sayfalar vs.) BM25 skorlarını olumsuz etkiledi.

7. Performans Değerlendirme Sonuçları

- 6 farklı sorgu için toplamda 60 dokümana manuel etiketleme yapıldı.
- Aşağıdaki metrikler hesaplandı:

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



8. Proje Sürecinden Öğrenilen Dersler

- Sadece içerik benzerliğine dayalı sıralama yetersiz kalabilir; bağlantı yapısı sıralama kalitesini ciddi biçimde etkileyebilir.
- Geniş çaplı veri kümeleriyle çalışırken ön filtreleme ve örnekleme stratejileri kritik rol oynar.
- Yapay metinler, tekrar eden sayfa yapıları ve HTML şablon kalabalığı, bilgi erişimini zorlaştırabilir. Bu durumlar için özel temizleme stratejileri geliştirmek gerekir.

9. Sonuç

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