

T.C.
İSKENDERUN TEKNİK ÜNİVERSİTESİ
MÜHENDİSLİK VE DOĞA BİLİMLERİ FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ

MÜHENDİSLİKTE BİLGİSAYAR UYGULAMALARI I
PROJE RAPORU
GÜZ DÖNEMİ - ARALIK, 2025

AKADEMİK KEŞİF VE ANALİZ PLATFORMU

HAZIRLAYAN
Oğuzhan Tutucu
Öğrenci No: 212523033

DERSİ VEREN
Dr. Ögr. Üyesi H. İbrahim OKUR

1. PROJE AMACI

Günümüzde akademik literatürün hızla büyümesi, araştırmacıların kendi alanlarındaki trendleri takip etmesini ve ilgili yayılara ulaşmasını zorlaştırmaktadır. Bu projenin temel amacı; araştırmacıların belirledikleri anahtar kelimeler üzerinden ArXiv veritabanını tarayarak, literatürdeki eğilimleri analiz eden, yazar iş birliklerini görselleştiren ve makalelerin "duygu durumunu" (Sentiment Analysis) ölçen web tabanlı bir karar destek sistemi geliştirmektir.

Ayrıca proje, akademik etiğin bir gereği olan kaynak gösterme işlemini kolaylaştmak adına, **Web Scraping** (Veri Kazıma) yöntemleri kullanarak ilgili makalelerin BibTeX formatındaki atıf verilerini otomatik olarak çekmeyi ve kullanıcıya sunmayı hedeflemektedir.

2. İŞ PAKETLERİ VE AÇIKLAMALARI

Proje süreci, çevik (agile) geliştirme yöntemleri benimsenerek aşağıdaki 5 temel iş paketine (WP) ayrılmıştır:

- **İP-1: Veri Madenciliği Altyapısı (Data Mining):** Python arxiv kütüphanesi kullanılarak API üzerinden verilerin (Başlık, Özeti, Yazar, Tarih, Kategori) çekilmesi ve Pandas DataFrame yapısında yapılandırılması.
- **İP-2: Görselleştirme Modülleri:** Çekilen verilerin Plotly ve Matplotlib kütüphaneleri kullanılarak Trend Analizi (Çizgi Grafik), Konu Modelleme (Kelime Bulutu) ve Yazar Ağı (Network Graph) grafiklerine dönüştürülmesi.
- **İP-3: Doğal Dil İşleme (NLP):** TextBlob kütüphanesi entegre edilerek makale özetlerinin pozitif/negatif duygu analizinin yapılması ve literatürün genel atmosferinin belirlenmesi.
- **İP-4: Web Scraping Modülü:** BeautifulSoup ve Requests kütüphaneleri kullanılarak, API'nin sağlamadığı BibTeX verilerinin ArXiv web sayfasından HTML parsing yöntemiyle kazınması.
- **İP-5: Arayüz ve Entegrasyon:** Tüm modüllerin Streamlit framework'ü ile birleştirilerek interaktif, karanlık mod (Dark Mode) uyumlu ve kullanıcı dostu bir web uygulaması haline getirilmesi.

3. YARARLANILAN TEKNOLOJİLER

Proje tamamen Python programlama dili kullanılarak geliştirilmiştir. Kullanılan temel kütüphaneler ve seçim nedenleri şunlardır:

- **Streamlit:** Web arayüzü (Frontend) hızlı ve etkili bir şekilde oluşturmak için kullanılmıştır. Session State yönetimi sayesinde veri kaybı olmadan sekmeler arası geçiş sağlanmıştır.
- **Arxiv API:** Veri kaynağı olarak, akademik camianın en güvenilir açık erişim platformu olduğu için tercih edilmiştir.
- **Pandas:** Çekilen verilerin tablo formatında işlenmesi, filtrelenmesi ve temizlenmesi için kullanılmıştır.
- **Plotly (Express & Graph Objects):** İnteraktif grafikler (Zoom/Pan yapılabilen) oluşturmak için seçilmiştir. Özellikle Ağ Analizi (Network Graph) kısmında statik grafiklerin yetersizliği nedeniyle Plotly'ye geçiş yapılmıştır.
- **TextBlob:** Hafif ve hızlı bir NLP kütüphanesi olduğu için makale özetlerinin duygusal analizinde (Sentiment Analysis) tercih edilmiştir.
- **BeautifulSoup4 & Requests:** Web Scraping işlemi için HTML yapısını parçalamak (Parsing) amacıyla kullanılmıştır.
- **NetworkX:** Yazarlar arasındaki karmaşık ilişkileri modellemek ve grafik teorisi algoritmalarını (Fruchterman-Reingold layout) uygulamak için kullanılmıştır.

4. ÖN YÜZ VE ARKA YÜZ GÖSTERİMLERİ

Uygulama, sol tarafta parametrelerin girildiği bir "Kenar Çubuğu" (Sidebar) ve sağ tarafta sonuçların sekmeler (Tabs) halinde gösterildiği "Ana Panel"den oluşmaktadır.

4.1. Ana Ekran ve Dashboard:

Kullanıcı anahtar kelimeyi girip "Analizi Başlat" butonuna bastığında, sistem arka planda API isteğini gerçekleştirir ve özet metrikleri (KPI) en üstte gösterir.

Toplam Makale

50

Farklı Yazar

154

En Aktif Yıl

2021

Kategori Sayısı

6

Trend Analizi

Konu Modelleme

Duygu Analizi

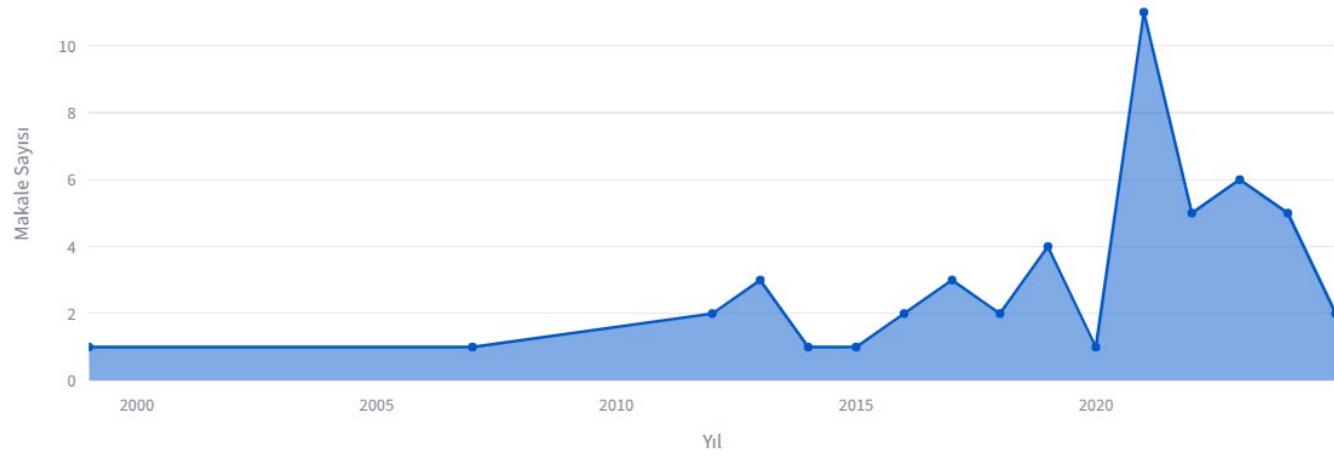
Yazar Ağrı

Detaylı Veri Seti

BibTeX Scraping

Zaman İçindeki Yayın Eğilimi

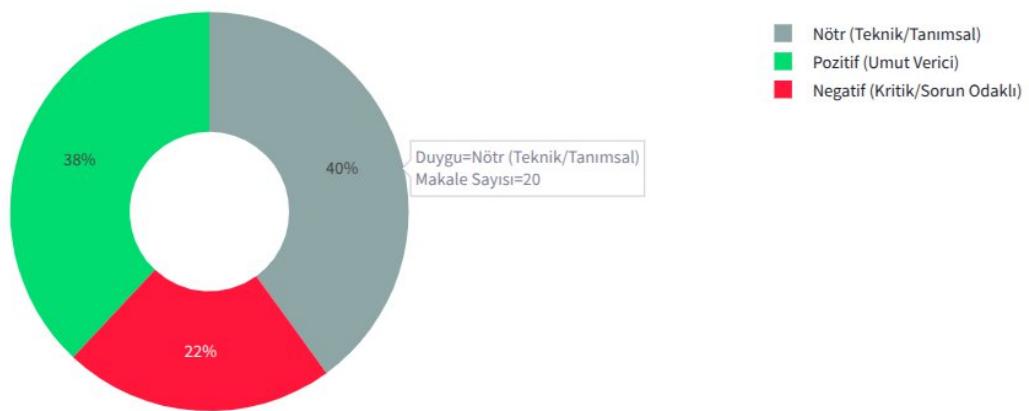
Yıllara Göre Yayın Trendi



4.2. Duygu Analizi ve NLP:

Sistem, makale özetlerini okuyarak literatürün ne kadarının "Umut verici (Pozitif)" ne kadarının "Sorun odaklı (Negatif)" olduğunu analiz eder.

Literatürün Duygu Durumu (Abstract Sentiment)

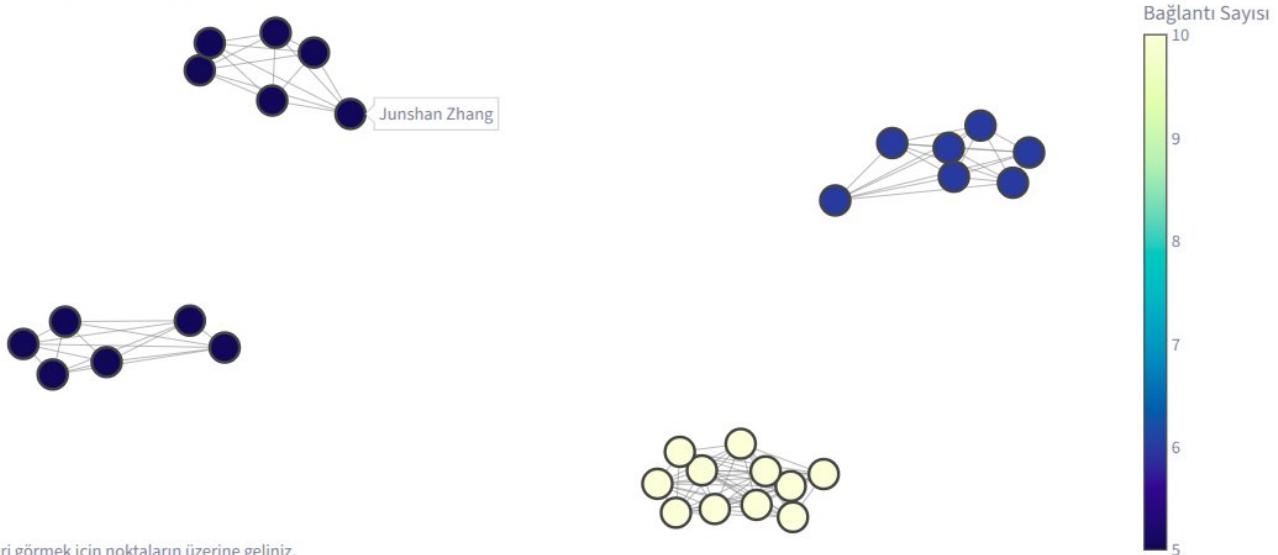


4.3. İnteraktif Yazar Ağlığı:

Yazarların iş birliği haritası, Plotly kullanılarak interaktif hale getirilmiştir. Kullanıcı fare ile düğümlerin üzerine gelerek yazar isimlerini görebilir.

Akademik İş Birliği Ağları (İnteraktif)

İşlemler için noktaların üzerine geliniz.



İsimleri görmek için noktaların üzerine geliniz.

5. KOD AÇIKLAMALARI

Projenin kritik fonksiyonları aşağıda açıklanmıştır:

- Veri Çekme Motoru (get_arxiv_data):
- Bu fonksiyon, kullanıcının girdiği anahtar kelimeye göre ArXiv sunucularına soru gönderir. Web Scraping şartını sağlamak adına, temel veri madenciliği burada API ile yapılırken, atıf verileri için ayrı bir scraping modülü yazılmıştır.
- Scraping Modülü (scrape_bibtex):
- Aşağıdaki kod bloğu, BeautifulSoup kütüphanesini kullanarak ArXiv web sayfasının HTML yapısını analiz eder. Özellikle Regex (Düzenli İfadeler) kullanılarak BibTeX formatındaki süslü parantezlerin {} dengesi kontrol edilir ve hatasız veri çekimi sağlanır.

```
def scrape_bibtex(paper_id):
    """BibTeX Scraper."""
    clean_id = re.sub(r'v\d+$', '', paper_id)
    url = f"https://export.arxiv.org/bibtex/{clean_id}"
    try:
        headers = {'User-Agent': 'Mozilla/5.0 (X11; Linux x86_64)'}
        response = requests.get(url, headers=headers, timeout=10)
        if response.status_code == 200:
            soup = BeautifulSoup(response.content, 'html.parser')
            bibtex_div = soup.find('div', id='bibtex')
            if bibtex_div: return bibtex_div.text.strip(), "Başarılı (Div Kaynağı)"
            text_content = soup.get_text()
            extracted_bib = extract_balanced_bibtex(text_content)
            if extracted_bib: return extracted_bib, "Başarılı (Metin Analizi)"
            return None, "Format bulunamadı."
        else: return None, f"Hata: {response.status_code}"
    except Exception as e: return None, f"Hata: {str(e)}"
```

- Ağ Görselleştirme (plot_optimized_network):
- Statik grafiklerin karmaşıklığı nedeniyle, NetworkX ile hesaplanan düğüm koordinatları Plotly Graph Objects yapısına aktarılmıştır. Bu sayede düğümlerin birbirini itmesi (Spring Layout) ve fare etkileşimi (Hover) sağlanmıştır.

6. KARŞILAŞILAN ZORLUKLAR VE ÇÖZÜMLER

Proje geliştirme sürecinde karşılaşılan teknik ve mantıksal zorluklar şunlardır:

- **Yazar Ağı Görselleştirmesi (The "Hairball" Problem):**
 - **Sorun:** Başlangıçta Matplotlib kullanıldığında, yazar isimleri ve bağlantı çizgileri üst üste binerek okunamaz bir "yumak" oluşturuyordu.
 - **Çözüm:** Görselleştirme kütüphanesi Plotly ile değiştirildi. İsimlerin sadece fare ile üzerine gelindiğinde (Hover) görünmesi sağlandı ve düğümler arası itme kuvveti (k parametresi) artırılarak grafik ferahlatıldı.
- **BibTeX Scraping ve Regex Hataları:**
 - **Sorun:** BibTeX formatı iç içe süslü parantezler içерdiği için basit Regex desenleri veriyi eksik çekiyordu (Örn: @misc etiketinin yarılmaması).
 - **Çözüm:** "Akıllı Parantez Sayacı" (Balanced Brackets) algoritması yazılarak, açılan her parantez için sayaç artırıldı, kapanan için azaltıldı. Böylece veri bloğunun tamamı hatasız çekildi.

7. SONUÇ

Bu proje ile, akademik araştırma süreçlerini hızlandıran, hem API hem de Web Scraping tekniklerini hibrit bir şekilde kullanan, modern arayüze sahip bir uygulama başarıyla geliştirilmiştir.