WEB İNDEXLEME UYGULAMASI

KOCAELİ ÜNİVERSİTESİ BİLGİSAYAR MÜHENDİSLİĞİ

YAZILIM LABORATUVARI PROJESİ

YASİN ÖMER KARA-FEYZA DEMİREL

180201077 – 190201110

y.omerkara1136@gmail.com feyzahae@gmail.com

ÖZET

Bizden istediğimiz programlama dilini kullanarak web indexleme uygulaması geliştirmemiz istendi. Bu uygulamada lokal bir web sitesi üzerinden kullanıcıdan url bilgisi alınacak ve daha sonra alınan bu bilgilerle birinci sayfada, sayfada geçen kelimeler ve kelimelerin frekanslarını yazdırmamız istendi.

İkinci sayfada ise verilen web sitesinin anahtar(en çok kullanılan kelimeler) kelimeleri çıkarmamız istendi.

Üçüncü sayfada ise iki web sitesi için anahtar kelimeler çıkarılacak ve bu anahtar kelimelere göre bir benzerlik formülü oluşturularak web sitelerinin benzerlik oranını oluşturmamız istendi.

İkinci ve üçüncü sayfaların sonuçları tek bir sayfada yazdırılacaktı.

Dördüncü sayfada ise bir url ve bir web sitesi kümesindeki her alt url ve bu urllerin de alt urlleri ile başta verilmiş olan ilk url arasında bir benzerlik skorlaması oluşturmamız istendi. Bu benzerlik skorlaması madde üçten daha farkı bir şekilde formülize edilecekti. Ve çıktı olarak her bir URL (bir web sitesi) için, sırasını, skorunu, alt URL'lerin ağaç yapısını ve her düğümdeki her bir anahtar kelimenin yer alma sayısı ile birlikte yazdırmamız istendi.

Beşinci ve son sayfada ise yinelemeli olarak 4. aşamadaki işlemleri tekrarlayarak benzer anahtar kelimeleri yazdırmamız istendi.

Son olarak çalışmamızı lokal bir web ortamında yayınlamamız istendi.

1.GİRİŞ

Projeyi Python programlama dili ile yazdık. Öncelikle Python'da local web sitesi nasıl oluşturulur bunu araştırarak local bi web sitesinde sitemizin dizaynını oluşturduk. Bunu yaptıktan sonra nasıl bir web sitesinden veri çekeriz diye araştırmalar yaptık. Bunun sonucunda BeautifulSoup kütüphanesini kullarak web sitesinden veri çektik. Daha sonrasında çektiğimiz verileri ayıklamamız gerekiyordu.

Bunun için öncelikle istediğimiz verilerin hangi etikette olduğunu bulup ona göre kodumuzu yazmaya başladık.

2. YÖNTEM

İlk sayfa için sayfadaki kelimeleri alıp bu kelimelerin frekansını tutmamız gerekiyordu. P etiketi içerisinde web sitesinin metinlerinin yazıldığını biliyorduk o yüzden p etiketi içerisindeki kelimeleri aldık ve bu kelimeler içersindeki sembolleri temizledik. '*,./()&…vb' gibi.

Sayfa 1:

Aldığımız bu kelimeleri sözlük yapısına aktardık ve frekanslarını bulduk. Daha sonrasında bu verileri oluşturduğumuz web sitesinin ilgili sayfasına yollayarak ekrana yazdırdık.

Daha sonra ikinci web sitesinde anahtar kelimeleri çıkarmamız gerekiyordu bunun için kelimelerin frekanslarını kullandık. Özellikle dikkate aldığımız kelimeler ise tittle ve description etiketlerinde bulunan kelimelerdi. Çünkü bu kelimeler sayfanın içeriğini belirlemede etkin rol oynuyordu.

Üçüncü sayfada ise iki web sitesinin birbirine benzerliklerini bulmamız gerekiyordu. Öncelikle ilk web sitesinin ve ikinci web sitesinin anahtar kelimelerini karşılaştırdık. Ortak anahtar kelime sayısını anahtar kelimesi daha fazla olan web sitesine böldük ve 100'le çarparak yüzdesel oran hesapladık.

Bu verileri ilgili web sayfamıza yollayarak oranı ve anahtar kelimeleri birlikte yazdırdık.

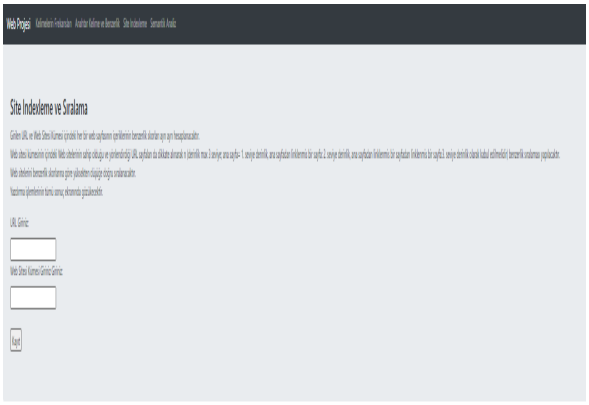
Sayfa 2-3:

Dördüncü kısımda ise öncelikle bir fonksiyonda web sitesinin içerisinde bulunan alt web sayfalarını bulduk. Bu kısımda web siteleri dışında da veriler elimize gelmişti ve öncelikle bu verileri temizledik. Aldığımız bu alt web sitelerini ise bir diziye attık. Bu dizideki her bir elemanı ise sözlük yapısından geçirerek anahtar kelimelerini bulduk daha sonra ise bu anahtar kelimeler ile ilk urlmizin anahtar kelimelerini karşılaştırarak benzerlik skorunu

hesapladık. Daha sonra tüm alt urller ve bu alt urllerin alt urlleri için bu işlemi tekrarladık.

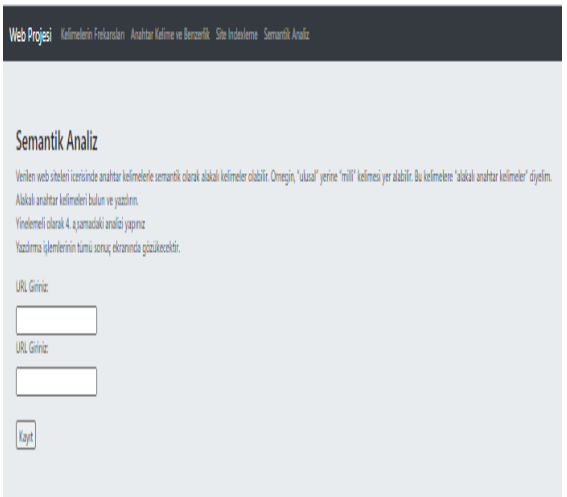
Sonuçları benzerlik skoruna göre sıralayarak web sitesine yazdırdık.

Sayfa 4:



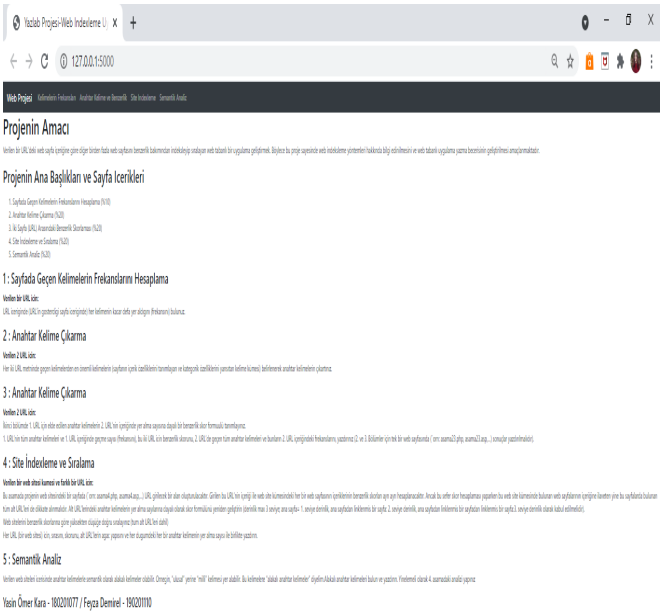
Beşinci kısımda ise benzer anahtar kelimeleri bularak örneğin ‘ulusal’ ve ‘milli’ bu kelimeleri yazdırdık. Burada yinelemeli olarak 4. Aşamayı tekrarlamamız gerekiyordu ve biz de 4. Aşamayı tekrarladık.

Sayfa 5:



Son olarak web sitemizin dizaynını güçlendirdik. Giriş kısmına her sayfayla ilgili detaylar yazarak kullanıcıyı bilgilendirdik. Ve her sayfanın amacını projenin amacını belirterek bir başlangıç sayfası oluşturduk.

Ana Sayfa:



3. DENEYSEL SONUÇLAR

Öncelikle farklı programlama dilleri kullanarak web siteleri oluşturmayı ve veri çekmeyi denedik ancak bu konuda en kolaylıkla projeyi sürdürdüğümüz dil Python oldu.

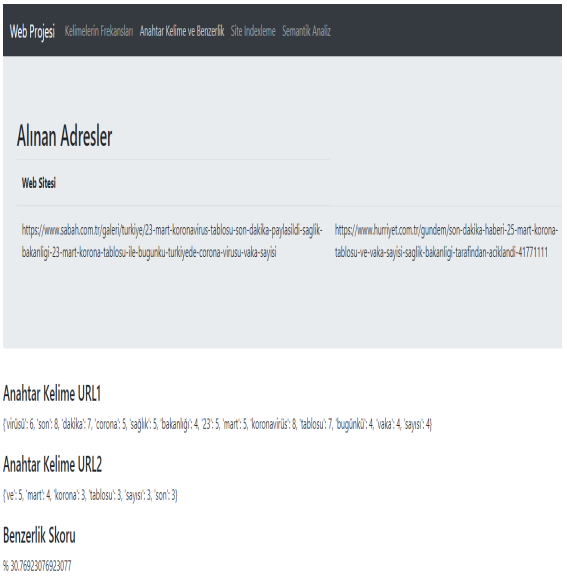
Kullanıcıdan url alıp bunu fonksiyonlarda kullanma konusunda başlangıçta sorun yaşadık ama daha sonrasında öncelikle veriyi bir html sayfasına yönlendirdik ve böylelikle bu sorunu çözmüş olduk.

Diğer sorun yaşadığımız şeylerden biri ise dördüncü sayfada web sitesinden url içeriklerini çekip bi dizide tutmaktı. Bu şekilde dizide tuttuğumuzda A etiketi içerisinde başka veriler de olduğu için urlleri ayıklamamız gerekiyordu. Diziden bu diğer verileri silmekte sorun yaşadık bu yüzden bu doğru verileri başka bir dizide tutarak bunu hallettik.

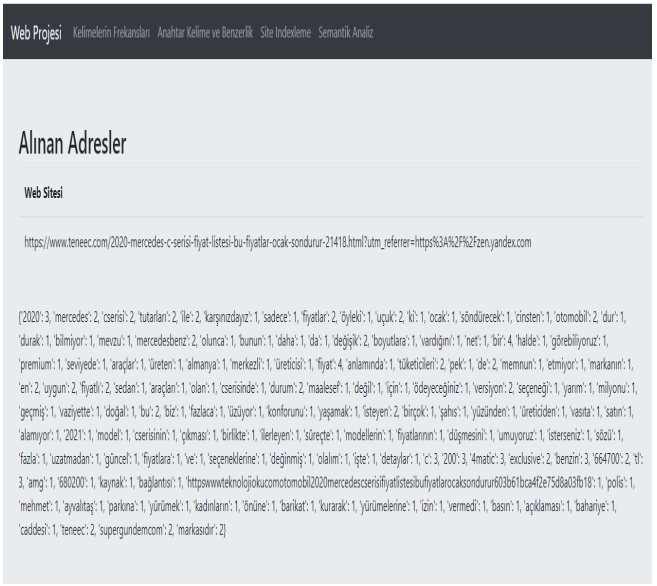
Eğer girilen değer bir web sitesi değilse ya da web sitesine erişim yoksa bu durum için de bir hata mesajı verisi yazdırdık.

Madde 4'te urllerin benzerliklerinin skorlamasını sıralayacaktık ve bunun için de sözlük yapısı kullanarak kolay bir sıralama elde ettik.

URL GİRİLDİĞİNDEKİ EKRAN ÇIKTISI:



Örnek Başka Bir Ekran Çıktısı:



4. KULLANDIĞIMIZ KÜTÜPHANELER

Projede kullandığımız kütüphaneler;

- Flask
- Bs4
- Operator
- Pip._vendor
- Wtforms

5. KAYNAKÇA

[1].Web Site

https://www.mobilhanem.com/flask-from-yapilari-get-post/

[2].Web Site

https://www.geeksforgeeks.org/extract-all-the-urls-from-the-webpage-using-python/

[3].Web Site

https://pythonspot.com/extract-links-from-webpage-beautifulsoup/

[4].Web Site

https://realpython.com/python-web-applications/

[5].Web Site

https://flask.palletsprojects.com/en/1.1.x/

6.AKIŞ DİAGRAMI