Kocaeli Üniversitesi

Bilgisayar Mühendisliği Bölümü

Yazılım Laboratuvarı II

Web Scraping Akademi Uyguaması

Oğuzhan Çelik-190202105

Projenin Özeti

Bu proje, Yazılım Laboratuvarı II dersi kapsamında geliştirilen "Web Scraping Akademi Uygulaması"dır. Uygulama, Google Scholar üzerinden belirli konularda arama yaparak makale bilgilerini toplar ve kullanıcıya sunar. Python diliyle geliştirilen uygulama, kullanıcıların belirlediği anahtar kelimeler doğrultusunda arama yaparak makale başlıklarını, yazar bilgilerini, yayımlanma tarihlerini ve alıntılanma sayısını çeker. Çekilen bu veriler MongoDB veritabanına kaydedilmekte ve Elasticsearch kullanılarak sorgulanabilmektedir.

Uygulama, web tabanlı bir arayüze sahip olup, kullanıcıların arama sonuçlarını filtrelemelerine ve makalelerin detaylarını görüntülemelerine olanak tanır. Ayrıca, yazım hataları tespit edilip düzeltilmiş öneriler sunularak kullanıcı deneyimi iyileştirilmiştir. Bu proje, web scraping, veri yönetimi ve sorgulama teknolojilerini bir araya getirerek, bilimsel makalelere hızlı ve etkili bir şekilde erişim sağlamayı amaçlamaktadır.

GİRİŞ

Web Scraping Akademi Uygulaması, kullanıcıların Google Scholar üzerinden belirli konular hakkında bilimsel makale aramaları yapmalarını ve bu makalelere ilişkin verileri çekmelerini sağlayan bir projedir. Python ile geliştirilen uygulama, kullanıcı dostu bir arayüzle birlikte, araştırmacıların ve öğrencilerin istedikleri konularda güncel bilimsel verilere kolayca ulaşmalarını hedeflemektedir. Kullanıcılar, arama yapmak istedikleri konu başlığını ve çekilecek sayfa sayısını girerek arama işlemini başlatabilirler.

Uygulama, çekilen verileri MongoDB veritabanına kaydetmekte ve Elasticsearch altyapısını kullanarak bu veriler üzerinde gelişmiş sorgulamalar yapılmasını sağlamaktadır. Veriler arasında makale başlıkları, yazar bilgileri, yayımlanma tarihleri ve alıntılanma sayıları gibi önemli bilgiler yer alır. Ayrıca, yazım yanlışlarına karşı kullanıcıya düzeltilmiş öneriler sunarak, doğru ve etkili sonuçlara ulaşmayı destekler.

Bu proje kapsamında geliştirilen Web Scraping Akademi Uygulaması, bilimsel araştırmalara hızlı ve kolay erişim imkânı sunarken, verilerin doğru bir şekilde saklanmasını ve analiz edilmesini sağlar. Yazılım Laboratuvarı II dersi kapsamında geliştirilen proje, araştırmacıların ve öğrencilerin ihtiyaçlarına uygun olarak tasarlanmış ve kullanıcıların arama işlemlerini optimize etmeyi amaçlamıştır.

1. TEMEL BİLGİLER

Projenin temelinde Python programlama dili ve Visual Studio Code geliştirme ortamı kullanılmıştır. Kodlar, **scraping.py** başlığı altında organize edilmiş olup, geliştirme sürecinde **requests**, **BeautifulSoup**,

pymongo ve **Elasticsearch** gibi önemli kütüphaneler kullanılmıştır. Bu kütüphaneler, web scraping işlemleri için HTTP istekleri, HTML içeriğinin analizi, MongoDB ve Elasticsearch ile etkileşim gibi işlevleri sağlamaktadır.

Kodlar, Google Scholar gibi bir web sitesinden bilimsel makale verilerini çekme işlemlerini gerçekleştirir. **requests** kütüphanesi ile web sitesine HTTP istekleri gönderilir, **BeautifulSoup** ile sayfa içeriği analiz edilerek makale başlıkları, yazar bilgileri, yayımlanma tarihleri, alıntılanma sayıları ve makale URL'leri gibi veriler elde edilir. Elde edilen bu veriler, hem MongoDB veritabanına kaydedilir hem de Elasticsearch ile sorgulanabilir hale getirilir.

Proje, Python'un esnekliği ve güçlü kütüphanelerinin sağladığı fonksiyonellikler sayesinde, bilimsel makalelerin toplanması ve analiz edilmesi süreçlerini kolaylaştırır. MongoDB ve Elasticsearch entegrasyonları, kullanıcıların çekilen veriler üzerinde detaylı sorgulamalar yapmasını mümkün kılarken, Web Scraping Akademi Uygulaması'nın işlevselliği ve verimliliğini artırmaktadır.

2. YÖNTEM

Bu projede, Python dili ile geliştirilen bir Web Scraping Akademi Uygulaması yapılmıştır. Kullanıcıların Google Scholar üzerinden bilimsel makale araması yapıp, bu makalelere dair bilgileri elde etmesi, verilerin MongoDB ve Elasticsearch gibi sistemlere kaydedilmesi sağlanmıştır. Projede kullanılan temel yöntemler ve teknolojiler aşağıda detaylandırılmıştır.

1. Web Scraping Süreci

Web scraping işlemleri için **requests** ve **BeautifulSoup** kütüphaneleri kullanılmıştır. **requests** kütüphanesi, Google Scholar'a HTTP istekleri göndermek ve sayfa içeriğini almak için kullanılmıştır. Ardından, **BeautifulSoup** kütüphanesi bu sayfa içeriğini analiz ederek istenen makale başlıkları, yazar adları, yayın yılları, makale URL'leri ve alıntılanma sayıları gibi bilgileri çekmiştir.

Scraping işlemini gerçekleştiren kod, **scraping.py** dosyasında **makaleleri_tara()** fonksiyonu ile organize edilmiştir. Bu fonksiyon, kullanıcıdan alınan arama sorgusunu Google Scholar üzerinde belirli sayıda sayfa boyunca arar ve her sayfadaki sonuçları analiz ederek veri listesini oluşturur. Elde edilen veriler daha sonra MongoDB'ye ve Elasticsearch'e kaydedilmek üzere düzenlenir.

2. Veri Kaydetme ve İşleme

Veriler, **MongoDB** ve **Elasticsearch** sistemlerine kaydedilerek ileride yapılacak sorgulamalar için kullanılabilir hale getirilmiştir. MongoDB, yapılandırılmamış verilerin kolayca saklanabilmesi için kullanılırken, Elasticsearch, veriler üzerinde hızlı aramalar ve filtrelemeler yapılmasını sağlamaktadır. Bu entegrasyon, kullanıcıların veri çekme işlemlerini sadece Google Scholar'dan yapmayıp, veritabanında daha önce saklanmış verilere de erişim sağlamaktadır.

Scraping işlemi sonrasında veriler MongoDB'ye **pymongo** kütüphanesi aracılığıyla kaydedilmektedir. Elasticsearch'e veri kaydetme işlemi ise **elasticsearch** kütüphanesi ile gerçekleştirilmiştir. **scraping.py** dosyasında, kullanıcı tarafından girilen her veri önce MongoDB'ye kaydedilmekte, daha sonra bu verilerin bir kopyası Elasticsearch'te indekslenmektedir.

3. Kullanıcı Arayüzü

Projede, kullanıcıların etkileşimde bulunabileceği iki temel HTML sayfası kullanılmıştır: **index.html** ve **sonuc.html**.

- **index.html**: Bu sayfa, kullanıcıların aramak istedikleri konuyu ve kaç sayfa boyunca veri çekmek istediklerini belirledikleri bir form içerir. Form verileri, Flask uygulaması aracılığıyla işlenir ve scraping işlemi başlatılır.

- **sonuc.html**: Arama sonuçlarının listelendiği sayfadır. Burada her makale için başlık, yazar, yayımlanma tarihi ve alıntılanma sayısı gibi bilgiler gösterilir. Ayrıca, kullanıcılar bu sayfada ilgili makalenin detaylarına ulaşmak için başlığa tıklayabilir ve makale detaylarını açılan bir dinamik pencerede (modal) görebilirler.

Ayrıca, makale PDF linkine ulaşılabiliyorsa, kullanıcıya PDF indir butonu da sunulmuştur.

4. Dinamik Arama ve Öneriler

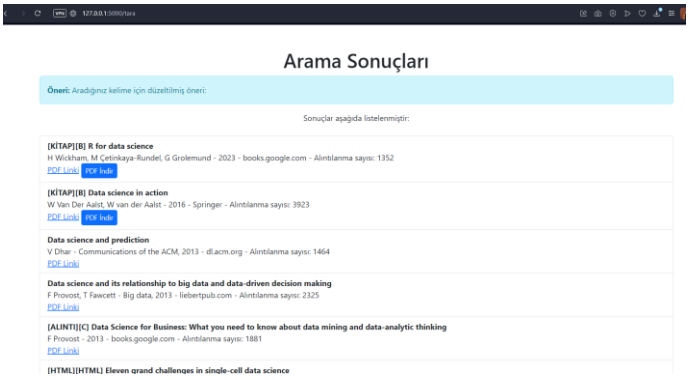
Projenin önemli bir özelliği, kullanıcının yanlış yazdığı arama sorgularını otomatik olarak düzeltme ve öneride bulunma kapasitesidir. Kullanıcı bir arama gerçekleştirdiğinde, Elasticsearch'ün **suggesters** özelliği sayesinde yazım yanlışları tespit edilmekte ve düzeltme önerileri kullanıcıya sunulmaktadır. Kullanıcı, hem düzeltilmiş önerileri görüp değerlendirebilir hem de aradığı veriye daha kolay ulaşabilir.

5. Verilerin Sunumu ve Modal Pencereler

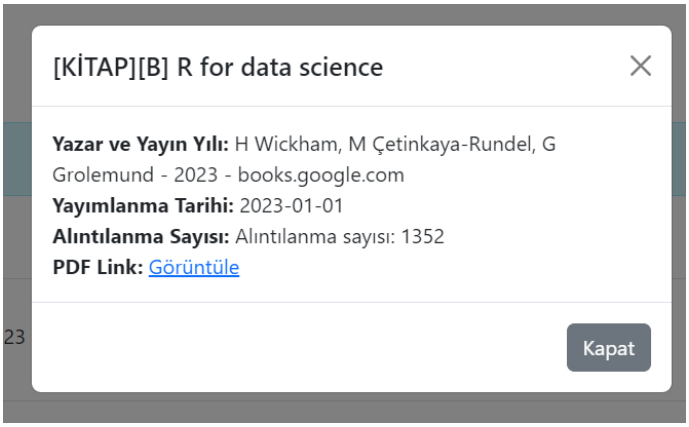
Kullanıcıların aradıkları makale detaylarına hızlı erişebilmesi için **sonuc.html** sayfasında makale başlıklarına tıklandığında açılan modal pencereler kullanılmıştır. Bu pencereler, veritabanından çekilen makale bilgilerini dinamik olarak kullanıcılara sunar. Flask ile oluşturulan backend yapısı, bu bilgilerin **/article_detail** route'u ile çekilmesini ve **jsonify** ile işlenmesini sağlar.

Bu yöntemlerle, kullanıcılar Google Scholar'dan kolayca veri çekebilir, elde edilen verileri analiz edebilir ve veritabanları aracılığıyla bu verilere her an erişim sağlayabilir.

Sonuc.html



Makale detay



3. SONUÇ

Bu proje kapsamında geliştirilen **Web Scraping Akademi Uygulaması**, Google Scholar üzerinden bilimsel makalelere yönelik veri toplama, bu verilerin MongoDB ve Elasticsearch gibi veritabanlarına kaydedilmesi ve kullanıcı dostu bir arayüz aracılığıyla bu verilere erişim sağlanmasını mümkün kılmıştır. Uygulama, Python'un güçlü kütüphanelerini kullanarak web scraping işlemlerini etkili bir şekilde gerçekleştirmiş ve veri analiz süreçlerinde kullanıcılara yardımcı olmuştur.

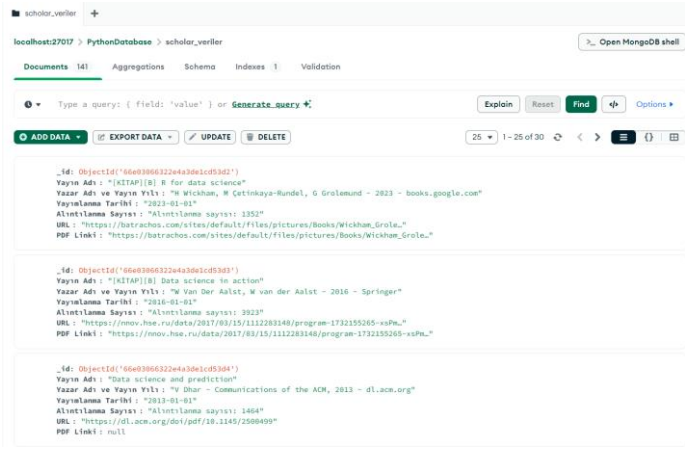
Proje, araştırmacılar ve öğrenciler için pratik bir çözüm sunarak bilimsel makalelerle ilgili verilere hızlı erişim sağlama hedefine ulaşmıştır. Uygulama, arama sorgularında yazım hatalarını düzeltebilme ve öneri sunma özellikleri ile kullanıcı deneyimini iyileştirmiştir. Ayrıca, makale bilgilerine ve detaylarına erişimi kolaylaştıran modal pencerelerle kullanıcı etkileşimi zenginleştirilmiştir.

Sonuç olarak, uygulama sayesinde bilimsel makalelere erişim süreci hızlandırılmış ve veri kaydetme, arama, filtreleme gibi işlemler sorunsuz bir şekilde gerçekleştirilmiştir. Projenin sağladığı bu başarılar, ileride benzer projelerde kullanılabilecek sağlam bir temel oluşturmuştur.

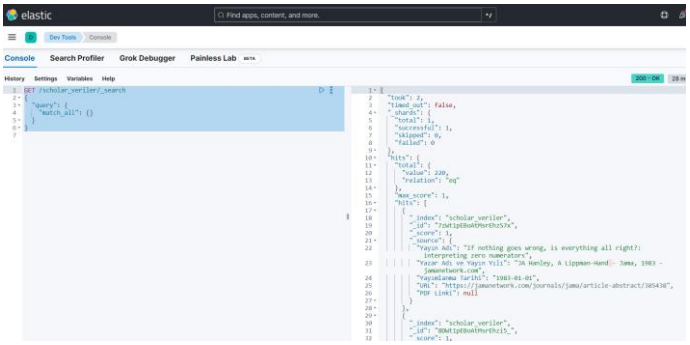
4. DENEYSEL SONUÇLAR

Index.html

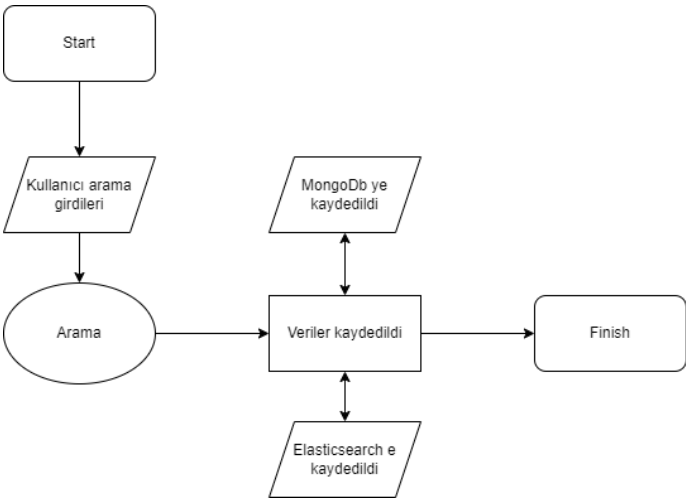
MongoDb



Elasticsearch



AKIŞ ŞEMASI



5. KAYNAKÇA

https://www.youtube.com/watch?v=8dTpNajx aH0

https://www.youtube.com/watch?v=JlHdv4Dfj q4

https://stackoverflow.com

https://www.youtube.com/watch?v=MH3641s 3Roc

https://codeahoy.com/index.html

https://www.youtube.com/watch?v=djfnjtYB2 co