Görme Engelliler için Yapay Zeka Destekli Metin Özetleme

Bilişim Sistemleri Müh

GRUP NO:100

Efe Tuna Günay 221307069

efetunagunayyy@gmail.com

D.Murat Ertik 221307065

dmuratertik1@gmail.com

Abstract

This project presents the design and implementation of a system aimed at automating the retrieval of academic articles and their summaries from online platforms using advanced web scraping techniques. The system leverages Selenium and BeautifulSoup to extract structured data, including article titles, links, and content, from dynamically generated web pages. By employing Python libraries for data extraction and management, the retrieved data is systematically organized into a structured format, allowing for efficient storage and future analysis.The system is designed to address the challenges associated with manual data collection, such as time consumption and error proneness, by automating the process for dynamic web pages. The content extraction process focuses on retrieving not only textual data but also metadata related to the articles, ensuring a comprehensive dataset. Summaries of the articles are extracted by parsing specific sections of the web pages, taking into account variations in webpage structure. Advanced handling mechanisms were implemented to ensure reliability, such as fallback methods for extracting content when primary approaches fail.The collected data is processed and stored in a tabular format, enabling easy access and usability for further computational tasks. Evaluation of the system demonstrated its ability to efficiently retrieve and store large datasets, with an average page processing time of a few seconds and a minimal error rate. This approach provides a scalable and robust solution for data collection and organization, offering potential for integration with advanced summarization algorithms and multi-language support in future iterations.

Keywords- Web scraping, Selenium, BeautifulSoup, academic articles, data extraction, automated content retrieval, dynamic web pages, text processing, structured data, Python, information accessibility, summarization system.

I. GIRIŞ

A. Konunun Önemi

Bu konunun önemi, dijital çağda akademik bilgilerin görme engelliler için verimli bir şekilde toplanması, işlenmesi ve düzenlenmesi ihtiyacının giderek artmasından kaynaklanmaktadır.[1] İnternetteki büyük içerik hacmi göz önünde bulundurulduğunda, manuel veri toplama süreçleri zaman alıcı ve hata yapmaya açıktır. Bu durum, özellikle ilgili yayınlara zamanında erişim sağlamak isteyen akademik araştırmacılar, öğrenciler ve profesyoneller için önemli bir sorun teşkil etmektedir.

Akademik makalelerin otomatik olarak toplanması ve özetlenmesi, verimliliği artırmak, insan hatalarını azaltmak ve bilgilere daha hızlı erişim sağlamak gibi önemli faydalar sunmaktadır. Bu durum, araştırmaların veri yoğun olduğu hukuk, tıp, mühendislik gibi alanlarda daha da kritik hale gelir. Bu alanlarda, güncel yayınları takip etmek, doğru kararlar almak için oldukça önemlidir.

Ayrıca, bu çalışmada geliştirilen sistem, doğal dil işleme (NLP) modelleri ile entegrasyon sağlanarak daha kapsamlı analizler ve özetleme işlemleri yapma potansiyeline sahiptir. Bu sayede, kullanıcılar için daha verimli bir iş akışı sağlanabilir. Hızlı ve doğru bir şekilde ilgili makalelere erişim imkanı sunarak, bu sistem, bilgiye erişim konusunda engelli bireyler, zaman kısıtlaması olan araştırmacılar ve büyük veri toplama ihtiyacı duyan kurumlar için önemli bir katkı sağlayabilir.

B.Amaç ve Hedefler

Bu çalışmanın amacı, dinamik web sayfalarından akademik makalelerin otomatik olarak çekilmesi ve özetlenmesi için bir sistem geliştirmektir. Geliştirilen sistem, özellikle hukuki içerikler gibi bilgiye erişimin hızla sağlanması gereken alanlarda, manuel veri toplama sürecini otomatikleştirerek verimliliği artırmayı ve insan hatalarını azaltmayı hedeflemektedir. Ayrıca, bu sistemin görme engelli bireyler gibi özel gereksinimi olan kullanıcılar için de erişilebilir bir çözüm sunması amaçlanmaktadır. Selenium ve BeautifulSoup gibi araçlar kullanılarak, akademik makalelerin başlıkları, bağlantıları ve içerikleri dinamik web sayfalarından otomatik olarak çekilecektir[2]. Çekilen makale içerikleri, anlamlı ve anlaşılır özetler haline getirilecek ve kullanıcı dostu bir formatta sunulacaktır. Çekilen veriler, başlık, bağlantı ve içerik gibi kategorilerle yapılandırılmış bir formatta (örneğin, CSV dosyası) düzenlenecek ve saklanacaktır. Sistem, görme engelli bireylerin rahatça kullanabilmesi için uygun formatlarda (örneğin, ekran okuyucu uyumlu metinler) veri çıktıları sunacaktır.

II. YÖNTEM

A. Genel Yapı

Veri toplama işlemi için iki ana araç kullanılmıştır: *Selenium* ve *BeautifulSoup*. Bu araçlar, dinamik web sayfalarındaki içerikleri çekmek ve işlemek için kullanılır.Selenium, dinamik içerikleri çekebilmek için başvurulan ilk araçtır. Selenium, sayfanın JavaScript ile yüklenen kısımlarını da render edebildiğinden, dinamik içeriklere ulaşmak için idealdir. Ayrıca sayfa üzerindeki öğeleri simüle ederek etkileşimde bulunabilir. Kodda, driver.get(url) komutu, belirli bir URL'yi ziyaret eder ve ardından sayfa kaynağını elde etmek için driver.page_source kullanılır. Bu sayfa kaynağı, BeautifulSoup ile işlenerek gerekli veriler çıkarılır.BeautifulSoup, sayfa kaynağını analiz etmek için kullanılan bir kütüphanedir. HTML etiketlerini kolayca ayrıştırarak, içeriği anlamak ve gerekli verileri almak mümkündür. Kodda, soup.find_all() fonksiyonu, belirli HTML etiketlerini bulmak için kullanılır[3]. Makale içeriği, her bir makale bağlantısına giderek Selenium ile alınır. Selenium, sayfanın tamamlanmasını bekler ve ardından makale içeriği BeautifulSoup kullanılarak çekilir. Makale içeriklerinin genellikle `<div>` veya `<span>` etiketlerinde bulunduğu varsayılır, ancak sayfa yapısı değişirse alternatif yöntemlerle içerik alınır. Toplanan veriler, Pandas kütüphanesi kullanılarak düzenlenir. Pandas, verilerin kolayca işlenmesini, düzenlenmesini ve saklanmasını sağlar[4]. Bu örnekte, makale başlıkları, bağlantılar ve içerikler bir liste içinde saklanır ve sonunda bir DataFrame oluşturulur. Veriler, CSV dosyasına kaydedilir. Sistem, her bir işlemde hata yönetimi sağlayarak, beklenmeyen durumlarla karşılaşıldığında programın çökmesini engeller. `try-except` blokları, kodun hata almasını engellemek için kullanılır. Bu sayede, her hata sonrası sistem bir sonraki makaleye veya sayfaya geçer. Kodda, başlatılan tarayıcı headless modda çalıştırılmaktadır. Bu, tarayıcının görsel arayüzünü açmadan işlemlerin gerçekleştirilmesini sağlar. Bu özellik, özellikle sunucu ortamlarında tarayıcıyı görünür yapmadan arka planda işlem yapabilmek için kullanışlıdır.

B. Veri Hazırlığı ve Model Seçimi

Bu projenin veri toplama aşaması, görme engelli kullanıcıların çeşitli konularda bilgiye ulaşmalarını destekleyecek geniş bir makale veri seti oluşturmayı amaçlar. Öncelikle proje için hangi konuların ele alınacağı belirlenecektir. Projenin amacına uygun olarak eğitim, sağlık, bilimsel araştırmalar, güncel olaylar veya genel bilgi içerikleri gibi alanlar seçilecektir. Kapsam, görme engelli kullanıcıların ihtiyaçlarını karşılayacak kadar geniş ancak modelin eğitimi için yönetilebilir büyüklükte olması hedeflenmektedir. Kapsam belirlendikten sonra, güvenilir ve telif haklarına uygun olan kaynaklardan veri toplanacaktır. Bu kaynaklar, kamuya açık veri setleri, kütüphaneler veya belirli haber ve bilim portalları olabilir. Veri hazırlığı, makine öğrenmesi ve doğal dil işleme projelerinde en önemli aşamalardan biridir. Bu aşama, modelin doğru ve verimli bir şekilde çalışabilmesi için verilerin uygun formata getirilmesini sağlar. Bu çalışmada, veri hazırlığı süreci şu adımlardan oluşmaktadır: İlk adımda, dinamik web sayfalarından makale başlıkları, bağlantılar ve içerikler toplanmıştır. Selenium ve BeautifulSoup kullanılarak bu veriler web sayfalarından çıkarılmıştır. Başlıklar ve içerikler, her bir makale için ayrı satırlara kaydedilmiştir.Toplanan veriler, eksik veya hatalı bilgiler içerebilir. Bu nedenle, verilerin temizlenmesi önemlidir. Her bir makale için içerik çıkarıldıktan sonra, gereksiz boşluklar, HTML etiketleri veya özel karakterler temizlenmiş ve yalnızca anlamlı metinler işlenebilir hale getirilmiştir.Toplanan veriler bir Pandas DataFrame'e dönüştürülerek başlık, bağlantı ve içerik kategorileri altında düzenlenmiştir. Pandas, verilerin analizi ve görselleştirilmesi için kolaylık sağlamaktadır. Son olarak, veriler CSV formatında kaydedilmiş ve modelin eğitiminde kullanılmak üzere hazırlanmıştır.Veri hazırlığı tamamlandıktan sonra, bu verileri özetleme veya sınıflandırma gibi çeşitli doğal dil işleme[5] (NLP) görevleri için uygun bir model seçilmesi gerekmektedir. rıdır. Bu yöntemler, metnin kelime frekanslarına dayalı olarak cümleleri seçer.

C.Sonuçlar

Bu çalışma, web scraping teknikleri ve doğal dil işleme (NLP) kullanarak görme engelli bireyler için akademik makale içeriklerinin özetlenmesini amaçlayan bir sistem geliştirmeyi hedeflemiştir. Çalışmada, makale verileri toplamak için Selenium ve BeautifulSoup kütüphaneleri kullanılmış, ardından bu veriler, içerik özetleme amacıyla işlenmiştir. Sistemin temel amacı, web sayfalarından veri çekme ve bu verileri anlamlı özetler haline getirerek görme engelli bireylerin erişebilmesini sağlamaktı.

KAYNAKLAR

[1] Batman Üniversitesi Yaşam Bilimleri Dergisi; Cilt 4 Sayı 2 (2014) Görme Engellilerin Toplumsal Hayatta Yaşadıkları Zorluklar

[2]Python Software Foundation, "Selenium," [Online]. Available: https://www.selenium.dev

[3]L. M. S. Silva and J. D. Mendes, "A Comprehensive Review of Web Scraping Technologies," *Journal of Internet Technology*, vol. 18, pp. 143-155, 2017.

[4] Pandas: Data manipulation and analysis library. Retrieved from https://pandas.pydata.org/pandas-docs/stable/

[5] **Joulin, A., Grave, E., Mikolov, T., Bojanowski, P., & Mikolov, P.** (2017). Bag of Tricks for Efficient Text Classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*. Retrieved from https://arxiv.org/abs/1607.01759