OKU Fen Bilimleri Enstitüsü Dergisi
8(3): 1405-1426, 2025

OKU Journal of The Institute of Science and Technology, 8(3): 1405-1426, 2025

Osmaniye Korkut Ata Üniversitesi Fen Bilimleri Enstitüsü Dergisi

Osmaniye Korkut Ata University Journal of The Institute of Science and Technology

Metin Özetleme ve Benzerlik Analizi Üzerine Prototip Bir Çalışma: Google Scholar Örneği

Onur ŞAHİN[1*], Rıdvan YAYLA[2]

[1]Bilecik Şeyh Edebali Üniversitesi, Lisansüstü Eğitim Enstitüsü, Bilgisayar Mühendisliği ABD Yüksek Lisans Programı, Bilecik, Türkiye
[2]Bilecik Şeyh Edebali Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, Bilecik, Türkiye

[1]https://orcid.org/0009-0000-8955-658X
[2]https://orcid.org/0000-0002-1105-9169
*Sorumlu yazar: onur2000sahin@outlook.com

Araştırma Makalesi

Makale Tarihçesi:
Geliş tarihi: 20.10.2024
Kabul tarihi:12.02.2025
Online Yayınlanma: 16.06.2025

Anahtar Kelimeler:
Doğal dil işleme
Benzerlik analizi
Metin özetleme
Berstum
LexRank
TextRank

ÖZ

Günümüzde bilgisayar destekli araçlar; akademik araştırmalarda bilgiye erişimi iyileştirme ve verimliliği optimize etmede önemli bir rol oynamaktadır. Bu bağlamda yapay zeka ve doğal dil işleme tekniklerinin kullanımı, araştırmacıların iş yükünü hafifletmekte ve daha hızlı sonuçlar elde etmelerini sağlamaktadır. Bu çalışmada; araştırma sürecinin özellikle literatür taraması ve kaynak bulma aşamalarında verimliliği artırmayı amaçlayan yeni bir prototip çalışma geliştirilmiştir. Bu çalışma, kullanıcıların girdiği anahtar kelimeler aracılığıyla Google Scholar platformundan belirlenen sayıda akademik makalenin PDF formatında otomatik olarak indirilmesini sağlamaktadır. Ardından Berstum, TextRank ve LexRank olmak üzere üç farklı özetleme algoritması seçeneği sunarak, indirilen makalelerin özetlerini çıkarır. Kullanıcı dostu bir arayüz aracılığıyla araştırmacılar istedikleri anahtar kelimeleri girebilir, benzerlik analizi için bir metin sağlayabilir ve tercih ettikleri özetleme algoritmasını seçebilirler. Elde edilen özetler ve benzerlik skorları arayüzde anlaşılır bir şekilde sunulur. İndirilen makalelerin içeriklerini kullanıcının girdiği metinle karşılaştırmak amacıyla metinlerdeki kelimelerin önem ve benzerliğini ölçen TF-IDF (Terim Frekansı-Ters Belge Frekansı) ve kosinüs benzerlik algoritmaları kullanılmıştır. Bu sayede, kullanıcının aradığı konularla ilgili makaleler ve ilgili bölümler tespit edilebilmektedir. Çalışmada ayrıca, geliştirilen prototipin ürettiği özetlerin kalitesini değerlendirmek için, özetlerin kullanıcıların girdiği referans metinlerle olan örtüşmesini, kesinliğini ve anlam bütünlüğünü ölçen Bleu, Rouge ve Meteor doğruluk metrikleri kullanılmıştır. Çalışmada üretilen özetlerde, çoğunlukla TextRank algoritmasının diğer algoritmalarla karşılaştırıldığında, bu doğruluk metriklerine göre büyük ölçüde %70 ila %94 aralığında çeşitli oranlarda daha yüksek doğruluk değerlerine ulaştığı tespit edilmiştir. Bu prototip çalışma, PDF dosyalarının çeşitli format ve düzenlerdeki yapısal farklılıklarını ele almak için çok aşamalı bir ön işleme süreci sunmaktadır. Bu sayede, farklı kaynaklardan gelen makalelerin özetleri ve benzerlik analizleri tutarlı bir şekilde gerçekleştirilebilmektedir.

A Prototype Study on Text Summarization and Similarity Analysis: Google Scholar Example

Research Article

ABSTRACT

Article History:
Received: 20.10.2024
Accepted: 12.02.2025
Published online:16.06.2025

Keywords:
Natural language processing
Similarity analysis
Text summarization
Berstum
LexRank
TextRank

Today, computer-aided tools play an important role in improving access to information and optimizing efficiency in academic research. In this context, the use of artificial intelligence and natural language processing techniques reduces the workload of researchers and enables them to obtain faster results. In this study, a new prototype study has been developed that aims to increase efficiency, especially in the literature review and source finding stages of the research process. This study automatically downloads a specified number of academic articles in PDF format from the Google Scholar platform through the keywords entered by users. Then, it provides summaries of the downloaded articles by offering three different summarization algorithm options: Berstum, TextRank and LexRank. Through a user-friendly interface, researchers can enter the desired keywords, provide a text for similarity analysis and select the summarization algorithm they prefer. The summaries and similarity scores obtained are presented in an understandable manner in the interface. In order to compare the content of the downloaded articles with the text entered by the user, TF-IDF (Term Frequency-Inverse Document Frequency) and cosine similarity algorithms, which measure the importance and similarity of words in the texts, were used. In this way, articles and relevant sections related to the topics searched by the user can be determined. In addition, Bleu, Rouge and Meteor accuracy metrics, which measure the overlap, precision and semantic integrity of the summaries with the reference texts entered by the users, were used in the study to evaluate the quality of the summaries produced by the developed prototype. In the summaries produced in the study, it was determined that the TextRank algorithm achieved significantly higher accuracy values in the range of 70% to 94% compared to other algorithms. This prototype study presents a multi-stage preprocessing process to address structural differences in PDF files in various formats and layouts. In this way, summaries and similarity analyses of articles from different sources can be performed consistently.

To Cite: Şahin O., Yayla R. Metin Özetleme ve Benzerlik Analizi Üzerine Prototip Bir Çalışma: Google Scholar Örneği. Osmaniye Korkut Ata Üniversitesi Fen Bilimleri Enstitüsü Dergisi 2025; 8(3): 1405-1426.

1. Giriş

Akademik araştırma süreçleri; çalışma konularının belirlenmesi, ilgili literatürün derinlemesine taranması, araştırma yöntemlerinin tasarlanması, veri toplama ve analiz aşamalarının planlanması, bulguların raporlanması ve sunulması gibi kapsamlı bir dizi adımdan oluşmaktadır (Creswell ve Creswell, 2018). Bu süreçler, özellikle literatür taraması ve bulguların yorumlanması evrelerinde, araştırmacıların büyük hacimli verilerle başa çıkarmayı gerektirmekte, zaman ve emek yoğun faaliyetler içermektedir (Boell ve Cecez-Kecmanovic, 2014). Bu bağlamda, doğal dil işleme, makine öğrenmesi ve metin madenciliği gibi bilgisayar bilimlerinin alt disiplinleri, akademik araştırma süreçlerinin verimliliğini artırma ve hızlandırma potansiyeline sahiptir. Son yıllarda, doğal dil işleme, makine öğrenmesi ve metin madenciliği gibi bilgisayar bilimlerinin alt disiplinleri, akademik araştırma süreçlerini daha verimli ve hızlı hale getirme konusunda önemli bir potansiyel sunmaktadır. Metin benzerliği tespiti, metin sınıflandırma, anahtar kelime çıkarma ve metin özetleme gibi NLP teknikleri, bu bağlamda araştırmacılara büyük kolaylık sağlamaktadır. (Mihalcea ve Radev, 2011). Özellikle derin öğrenme modelleri, daha karmaşık ve bağlamsal metin analizleri yapma olanağı tanıyarak metin özetleme ve benzerlik analizi gibi görevlerde kayda değer ilerlemeler sunmuştur (Ghanem ve ark., 2024). 2024'te geliştirilen EYEGLAXS (Easy Yet Efficient larGe LAnguage model for eXtractive

Summarization) gibi çerçeveler, büyük dil modellerini kullanarak uzun metinlerin özetlenmesinde hem doğruluk hem de verimlilik açısından önemli ilerlemeler sağlamıştır (Hemamou ve Debiane, 2024).

Metin benzerliği tespiti, araştırmacıların çalışma konularıyla ilgili kaynakları bulmalarına yardımcı olurken metin özetleme ve benzerlik analizi alanında, literatürde çeşitli algoritmalar yaygın olarak kullanılmaktadır. Özellikle TextRank ve LexRank, grafik tabanlı yaklaşımlarıyla öne çıkan özetleme yöntemleridir (Mihalcea ve Tarau, 2004; Erkan ve Radev, 2004). Bertsum ise, BERT tabanlı derin öğrenme modellerinin metin özetleme görevine uyarlanmasıyla son yıllarda dikkat çeken bir yöntem olmuştur (Liu ve Lapata, 2019). Bu çalışmada, söz konusu algoritmaların performansı karşılaştırmalı olarak incelenmiş ve literatür taraması sürecine entegre edilmiştir. Bu algoritmalar, uzun metinlerin ana fikirlerinin hızlı bir şekilde yakalanmasını sağlayarak literatür taraması süreçlerini önemli ölçüde kolaylaştırmaktadır (Nenkova ve McKeown, 2011). Bu algoritmaların avantajlarından yararlanarak, bu çalışmada Python programlama dili kullanılarak, bahsi geçen zorlukları ele almak ve araştırmacılara daha hızlı ve etkili bir literatür taraması deneyimi sunmak için Google Scholar temelinde PDF dokümanlarından metin çıkarma, TF-IDF tabanlı benzerlik analizi ve gelişmiş özetleme algoritmalarını bir araya getiren prototip bir çalışma önerilmiştir.

Google Scholar platformu, akademik araştırmalarda yaygın olarak kullanılan ve geniş bir kapsama sahip bir arama motoru olarak öne çıkmaktadır (Gusenbauer ve Haddaway, 2020). Geliştirilen prototip, kullanıcıların girdiği anahtar kelimeler ve benzerlik analizi yapılacak metinler aracılığıyla Google Scholar platformundan kullanıcının belirlediği sayıda akademik makaleyi PDF formatında otomatik olarak indirmektedir. (Beel ve ark., 2016) Bu işlem için "Requests" kütüphanesi kullanılmaktadır. Belirlenen anahtar kelimeler ile Google Scholar platformu üzerinden PDF formatındaki ilgili makaleler toplanmakta ve "PDFPlumber" kütüphanesi kullanılarak bu makalelerin metinleri çıkarılmaktadır. Ayrıca, Term frequency-inverse document frequency (TF-IDF) tabanlı bir cümle benzerlik ölçütü kullanılarak kullanıcı tarafından girilen metin ile makale metinleri arasındaki benzerlikler hesaplanmaktadır. Tf-idf, metin madenciliğinde yaygın olarak kullanılan bir vektör uzay modeli tekniğidir ve cümlelerin sözcük frekanslarına ve sözcüklerin dokümanlardaki dağılımlarına dayalı olarak benzerliklerini ölçmektedir (Ramos, 2003). Kullanıcı tarafından belirlenen benzerlik eşiğinin üzerinde olan cümleler, benzerlik oranlarıyla birlikte raporlanmakta ve olası kaynak metinler kullanıcıya sunulmaktadır.

Çalışmanın diğer bir işlevi metin özetlemedir. Bu çalışmada kullanıcılarına Bertsum, TextRank ve LexRank olmak üzere üç farklı özetleme algoritması seçenek olarak sunulmaktadır. Kullanıcılar, bir Python kütüphanesi olan “PyQt5” ile geliştirilmiş kullanıcı arayüzü üzerinden istedikleri özetleme algoritmasını seçebilmekte, seçilen algoritma ile oluşturulan özetler ise ayrı dosyalara kaydedilmektedir. Google Scholar platformundan toplanan ve PDF formatında indirilen makaleler, öncelikle bir dizi ön işlemden geçirilmektedir. Python programlama dili temelinde gelişmiş bir doğal dil işleme kütüphanesi olan Natural Language Toolkit (NLTK) kullanılarak yapılan ön işlemler, metinlerin küçük harfe dönüştürülmesi, durak kelimelerin kaldırılması, sayfa numarası, dergi no vb. anlam bütünlüğünü

bozacak sayısal değerlerin temizlenmesi gibi çeşitli işlemleri içermektedir (Bird ve ark., 2009). Ön işlemden geçirilen metinler; Bertsum (Liu ve Lapata, 2019), TextRank (Mihalcea ve Tarau, 2004) ve LexRank (Erkan ve Radev, 2004) olmak üzere üç farklı özetleme algoritması kullanılarak özetlenmektedir.

Prototipin sunduğu özetleme algoritmalarının başarımını değerlendirmek ve birbirleriyle karşılaştırmak amacıyla, ayrı bir çalışma kapsamında Meteor (Banerjee ve Lavie, 2005), Rouge (Lin, 2004), Bleu (Papineni ve ark., 2002) gibi yaygın olarak kullanılan metin özetleme başarı metrikleri kullanılmıştır. Bu metrikler kullanılarak elde edilen sonuçlar, özetleme algoritmalarının etkinliğini değerlendirmede önemli bir dayanak oluşturmaktadır. Bu sayede araştırmacılar, çalışma konularıyla ilgili literatürdeki olası kaynak metinleri sistematik bir şekilde tespit edebilmekte, farklı özetleme algoritmaları tarafından oluşturulan özetleri karşılaştırabilmekte ve araştırma süreçlerini daha verimli yürütebilmektedirler. Prototip, Google Scholar platformu üzerinden akademik kaynaklara erişim sağlayarak, PDF dosyalarını otomatik olarak işleyerek ve farklı özetleme algoritmaları sunarak araştırmacılara literatür taraması ve analiz süreçlerinde kolaylık sağlamaktadır. TF-IDF tabanlı benzerlik analizi ise, kullanıcının ilgi alanına giren makaleleri ve makaleler içerisindeki ilgili bölümleri hızlı bir şekilde tespit etmesine yardımcı olmaktadır. Geliştirilen prototip çalışmanın, bilgisayar destekli metin analizi tekniklerinin akademik araştırma süreçlerine entegrasyonunu sağlaması ve bu alana katkı sunması öngörülmektedir.

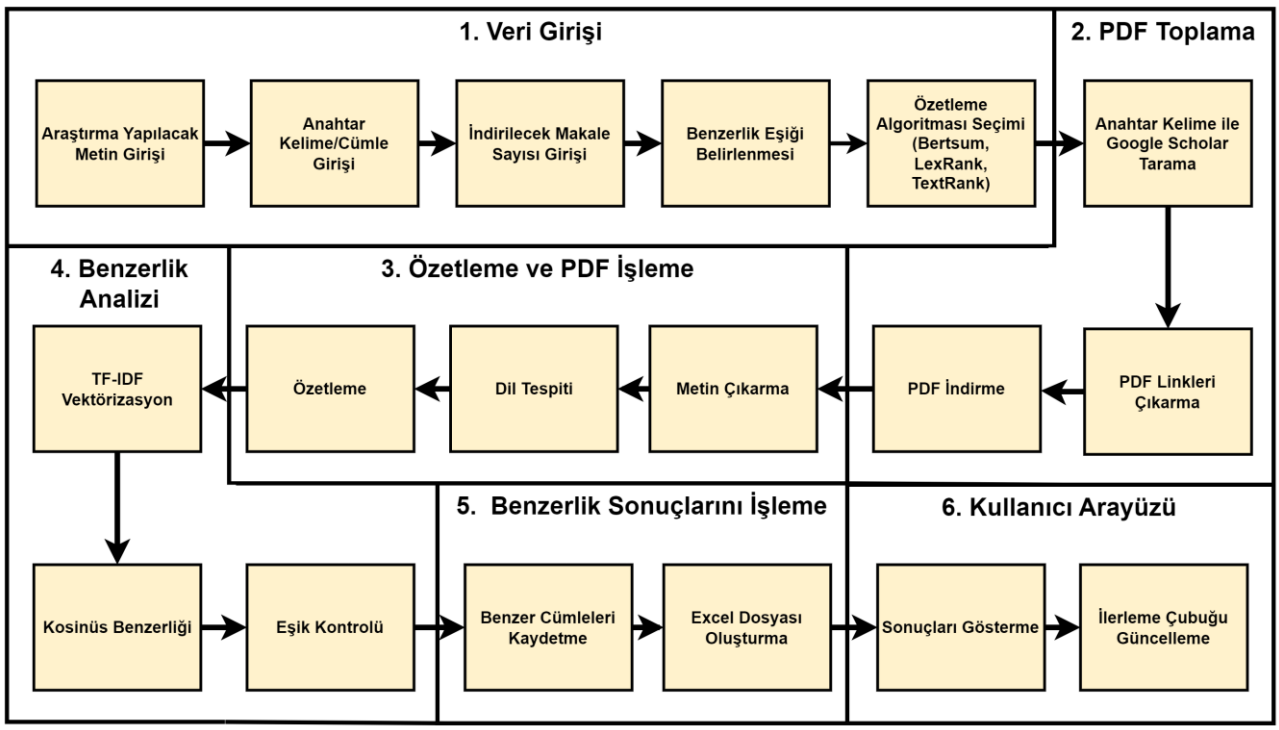
Akademik araştırma süreçlerinde, bilgisayar destekli teknolojilerin rolü ve bu teknolojilerin literatür taraması ile metin özetleme işlevlerine entegrasyonu son yıllarda önemli bir araştırma alanı haline gelmiştir. Doğal dil işleme ve makine öğrenmesi tekniklerini kullanan LiterEV gibi araçların geliştirilmesi; literatür incelemelerini ve metin özetlemeyi otomatikleştirerek akademik araştırma süreçlerinde önemli bir rol oynayarak, bu alana katkı sağlamaktadır (Orel ve ark., 2023). Metin benzerliği tespiti ve kaynak bulma konusunda, Alzahrani ve ark. (2011) tarafından yapılan başka bir çalışmada, n-gram tabanlı bir metin benzerliği ölçüm tekniği önerilmiş ve bu teknik akademik makalelerde intihal tespiti için kullanılmıştır. Ancak bu çalışmada, metin benzerliği tespiti daha çok intihal kontrolü amacıyla geliştirilmiş, araştırma süreçlerindeki literatür taraması bağlamında ele alınmamıştır (Alzahrani ve ark., 2011). Literatür taramasında metin özetleme tekniklerinin kullanımına ilişkin çalışmalar da mevcuttur. See ve ark. (2017) çalışmasında, Pointer-Generator Network modeliyle metin özetlemede bilgi kaybı sorununu ele almışlardır. Çalışmada, bir dokümanın önemli kısımları Pointer-Generator Network modeliyle birleştirilir ve uzun metinlerin özetlenmesinde etkili sonuçlar sunar (See ve ark., 2017). Benzer bir çalışmada, Cohan ve ark. (2018) ise uzun bilimsel metinlerin özetlenmesi için hiyerarşik bir dikkat mekanizması önermişlerdir (Cohan ve ark., 2018). Bu tür çalışmalar, metin özetleme tekniklerinin araştırma süreçlerinde kullanımına ışık tutsa da metin benzerliği tespiti ile entegrasyonu ele alınmamıştır.

Metin benzerliği ve özetleme işlevlerini bir araya getiren az sayıda çalışma mevcuttur. 2024'te yayımlanan bir çalışmada, LLM (Büyük Dil Modelleri) ve NLP tekniklerini birleştiren otomatik literatür tarama sistemleri, RAG (Retrieval-Augmented Generation) mimarisiyle PDF dosyalarından doğrudan

literatür özetleri üretmiş ve bu süreçte GPT-3.5-turbo gibi modellerin ROUGE-1 skorunda %36.4 performans sağladığı gösterilmiştir. Bu tür sistemler, literatür taramasının hızını artırırken, benzerlik analizi ve özetleme entegrasyonunu da optimize etmektedir (Fateh Ali ve ark., 2024). Başka bir çalışmada Cao ve arkadaşları Tf-idf tabanlı bir metin benzerlik ölçümü ile Bert tabanlı bir özetleme modelini birleştirerek bir literatür inceleme aracı geliştirmişlerdir (Cao ve ark., 2018). Bununla birlikte, bu araç pdf dokümanının otomatik işlenmesini ve Google Scholar platformu üzerinden kaynak tespitini kapsamamaktadır. Bu çalışma, Google Scholar platformundaki pdf uzantılı kaynakların taranarak hem özetlenmesini hem de kullanıcının yazmış olduğu referans metin ile kaynaklardaki benzerliğin ölçülmesini sağlayarak literatürdeki bu boşluğu doldurmayı hedeflemektedir. Prototip çalışma, Tf-idf tabanlı metin benzerliği ölçümü, Google Scholar entegrasyonu, PDF dokümanlarından metin çıkarma ve Bertsum tabanlı özetleme işlevlerini bir araya getirerek kapsamlı bir analiz sunmaktadır. Böylece araştırmacılar, tek bir uygulama üzerinden kaynak tespiti, içerik analizi ve özetleme işlemlerini gerçekleştirebilmektedirler.

2. Materyal ve Metot

Veri toplama, pdf işleme, benzerlik hesaplama ve özetleme süreçleri ve uygulamanın çalışma prensibi Şekil 1'de gösterilmiştir.



Şekil 1. Çalışma prensibi

2.1. Veri Toplama

Veri toplama süreci; çalışmanın başlangıç noktasını oluşturur ve doğru kaynaklara ulaşmak için kritik bir öneme sahiptir. Özellikle akademik araştırmalarda, veri toplama sürecinin sistematik ve kapsamlı olması, çalışmanın güvenilirliği ve geçerliliği açısından büyük önem taşır.

Bu çalışmada, kullanıcı tarafından girilen metin, ilgili literatür taraması için temel oluşturmaktadır. Bu metin, çalışmada referans metin olarak adlandırılacaktır. Referans metin, kullanıcı tarafından intihal kontrolü yapılmak istenen metni temsil eder. Kullanıcı arayüzü aracılığıyla girilen bu metin, daha sonraki aşamalarda kullanılmak üzere programa kaydedilir. Ayrıca kullanıcı, araştırmak istediği konu ile ilgili anahtar kelimeleri ve incelenecek makale sayısını da sisteme girer.

Çalışmada, belirtilen anahtar kelimeler ve makale sayısı doğrultusunda Google Scholar akademik arama motoru kullanılarak ilgili makalelerin PDF formatında toplanması işlemi otomatik olarak gerçekleştirilmektedir. Bu amaçla, Python programlama dili bünyesindeki Selenium kütüphanesi kullanılarak Google Scholar platformunda otomatik tarama ve veri toplama işlemleri gerçekleştirilmiştir. Selenium, web sayfalarını simüle ederek dinamik içerikleri işleyebilme yeteneği sayesinde Google Scholar gibi platformlardan veri toplamak için etkili bir araçtır (Naing ve ark., 2023). Google Scholar platformunda belirtilen anahtar kelimeler ile arama yapılır ve veri madenciliği işlemi ile belirtilen sayıda PDF dokümanı elde edilene kadar ilgili anahtar kelimeleri içeren tüm sayfalar kontrol edilir. Arama sonuçlarından elde edilen PDF erişim linkleri toplanır ve daha sonraki aşamalarda kullanılmak üzere kaydedilir. Kullanıcı, indirilen makalelerin referans metinle olan benzerliğini değerlendirirken kullanılacak bir benzerlik eşiği belirleyebilir. Bu eşik, TF-IDF ve kosinüs benzerliği hesaplamaları sonucunda elde edilen benzerlik skorlarının karşılaştırılacağı bir sınır değerini temsil eder.

2.2. Kaynakların İndirilmesi ve İşlenmesi

Veri toplama aşamasında elde edilen PDF dokümanlarının indirilmesi ve işlenmesi, çalışmanın ikinci aşamasını oluşturmaktadır. Bu aşamada, Google Scholar platformundan toplanan PDF erişim linkleri kullanılarak ilgili dokümanlar indirilir ve metin madenciliği işlemlerine tabi tutulur. Bu işlemler için Python'da Requests ve PDFPlumber gibi çeşitli kütüphaneler kullanılmıştır. İlk olarak, Requests kütüphanesi kullanılarak, toplanan PDF dosyaları Google Scholar platformundan indirilir ve yerel diske kaydedilir. Daha sonra, PDFPlumber kütüphanesi devreye girer ve indirilen her bir PDF dosyası işlenir. PDFPlumber, PDF dosyalarını okuyup analiz etmeyi sağlayan güçlü bir kütüphanedir. Metin çıkarma işlemi, makalenin giriş bölümünden başlayarak kaynakça bölümüne kadar olan kısmı kapsamaktadır. Bu amaçla, "Giriş", "Introduction" gibi yaygın olarak kullanılan ve giriş bölümünü ifade eden kelimeler ile metin çıkarma işlemine başlanır. "Kaynak", "Kaynakça", "References" gibi kaynakça bölümünü belirten kelimeler tespit edildiğinde ise metin çıkarma işlemi durdurulur. Eğer bu başlangıç ve bitiş kelimeleri PDF dosyasında bulunamazsa, ilk ve son iki sayfa otomatik olarak çıkartılır ve kalan kısımdan metin çıkarma işlemi gerçekleştirilir. Metin çıkarma işleminin ardından, analizlerin doğruluğunu ve verimliliğini artırmak için bir dizi ön işleme adımı uygulanır. Öncelikle, parantez içerikleri ve gereksiz karakterler temizlenerek metin daha sade ve anlaşılır hale getirilir. Daha sonra, metinlerde büyük-küçük harf ayrımından kaynaklanan farklılıkları ortadan kaldırmak ve analizleri tutarlı hale getirmek için tüm metinler küçük harfe dönüştürülür. Sayfa numarası, doi numarası vb.

sayısal değerler genellikle metin özetleme ve benzerlik analizi için önemli bilgiler içermediğinden ve gereksiz gürültü oluşturabileceğinden metinlerden temizlenir. Benzer şekilde, "Tablo", "Şekil", "Figure" ve "Görsel" gibi ifadeler içeren satırlar da genellikle tablo ve şekil başlıklarını temsil eder ve metin özetleme ve benzerlik analizi için gereksiz bilgiler içerir. Bu nedenle, bu satırlar da anlam akışını bozmamak ve analizlerin doğruluğunu artırmak için çıkartılır, bu sayede yalnızca makalenin temel metin içeriği korunmuş olur. Bu aşamadan sonra, makalelerin dili "langdetect" kütüphanesi kullanılarak otomatik olarak tespit edilir. Dil tespiti, doğal dil işleme uygulamalarında önemli bir adımdır çünkü farklı dillerdeki metinler farklı dilbilgisi kurallarına ve sözcük yapılarına sahiptir. Dil tespitinin ardından, NLTK kütüphanesi kullanılarak metin tokenlere (kelime veya cümle) ayrılır ve tespit edilen dile uygun durak kelimeler (stopwords) metinden çıkarılır (Loper ve Bird, 2002). Bu sayede, daha sonraki aşamalarda dile özgü özetleme ve analiz teknikleri kullanılabilir.

2.3. Tf-idf Tabanlı Benzerlik Hesaplama

Metin özetleme işleminden sonra, özetlenen metin ile referans metin arasındaki benzerlikleri tespit etmek için TF-IDF (Term Frequency-Inverse Document Frequency) tabanlı benzerlik hesaplama yöntemi kullanılmıştır. TF-IDF, bir belgedeki terimlerin önemini ölçen bir istatistiksel yöntemdir ve metin madenciliği ile bilgi erişiminde yaygın olarak kullanılmaktadır (Mohd Sofi ve Selamat, 2023; Setiawan ve Adnyana, 2023; Thirumahal, 2024). TF-IDF, iki temel bileşene dayanır: Terim Frekansı (TF): Bir terimin belirli bir belgede veya cümlede kaç kez geçtiğini gösterir. Bu, terimin o belge içindeki göreceli önemini yansıtır. Ters Belge Frekansı (IDF): Bir terimin tüm belge koleksiyonunda ne kadar nadir veya yaygın olduğunu ölçer. Nadir görünen terimler, belgeleri birbirinden ayırmada daha etkili oldukları için daha yüksek IDF değerlerine sahip olurlar.

Bir terimin TF-IDF skoru, bu iki bileşenin çarpımı ile hesaplanır ve Denklem 1'deki gibi hesaplanır.

$$TF - IDF(t, d) = TF(t, d) \times IDF(t) \quad (1)$$

Bu formülde, t belirli bir terimi, d ise belgeyi temsil eder.

Çalışmada, TF-IDF yöntemi kullanılarak referans metin ile özetlenen metinler arasındaki cümle düzeyinde benzerlik hesaplaması yapılmıştır. İlk olarak, hem kullanıcının girmiş olduğu metin hem de işlenmiş makale metinleri cümlelere ayrılır. Daha sonra, her bir cümle için Python programlama dilindeki "TfidfVectorizer" sınıfı kullanılarak TF-IDF vektörleri oluşturulur. Bu vektörler, cümlelerin terim ağırlıklarına dayalı olarak matematiksel bir temsilini sunar. Ardından, referans metindeki her bir cümle ile özetlenen metindeki her bir cümle arasında kosinüs benzerliği hesaplanır (Fauzi ve ark., 2022; Januzaj ve Luma, 2022). Kosinüs benzerliği, iki vektör arasındaki açının kosinüsünü hesaplayarak benzerliklerini ölçer. Sonuç olarak, 0 ile 1 arasında bir benzerlik skoru elde edilir; 1 değeri mükemmel bir benzerliği, 0 değeri ise hiçbir benzerlik olmadığını gösterir. Bu adım, cümleler arasındaki semantik

benzerliği belirlemede kritik bir adımdır. Uygulamada, kullanıcıya benzerlik eşiği belirleme seçeneği sunulmaktadır. Bu eşik değeri, referans metin ile özetlenen metin arasındaki kabul edilebilir benzerlik düzeyini belirler. Eşik değeri ne kadar yüksek olursa, tespit edilen benzerlikler o kadar yüksek olur. Ancak, çok yüksek bir eşik değeri, bazı anlamlı benzerliklerin gözden kaçırılmasına neden olabilir. Önerilen başlangıç eşiği değeri %75'tir. Ancak, kullanıcılar kendi ihtiyaçlarına ve analiz ettikleri metinlerin özelliklerine göre bu değeri ayarlayabilirler. Eşik değeri aşan skorlar, kullanıcının girmiş olduğu metin ile özetlenen metin arasında anlamlı bir benzerlik olduğunu gösterir ve bu sonuçlar hem ekranda görüntülenir hem de bir Excel dosyasına kaydedilir.

2.4. Metin Özeti Oluşturma

Bu çalışmada, indirilen PDF dosyalarından çıkarılan metinlerin özetlerini oluşturmak için Bertsum, LexRank ve TextRank olmak üzere üç farklı özetleme algoritması kullanılmıştır. Bu algoritmalar, metin özetleme problemini farklı yaklaşımlarla ele alarak çeşitli avantajlar sunmaktadır.

Bertsum, Bidirectional Encoder Representations from Transformers (BERT) mimarisine dayanan ve metin özetleme görevine özel olarak uyarlanmış bir dil modelidir (Liu ve Lapata, 2019). BERT'in temelini oluşturan Transformer mimarisi, dikkat mekanizmaları aracılığıyla metinlerdeki uzun mesafeli bağımlılıkları yakalayabilen derin öğrenme tabanlı bir modeldir (Vaswani ve ark., 2017). Bertsum, bu mimariyi kullanarak metni izole cümleler şeklinde değil, genel bağlamı içinde analiz eder. Bu sayede, geleneksel n-gram tabanlı istatistiksel yöntemlerin aksine, metnin anlamsal yapısını daha bütüncül bir şekilde kavrayarak daha kaliteli özetler üretir (Devlin ve ark., 2018). Model, çıkarıcı özetleme (extractive summarization) tekniğini benimseyerek özet için en önemli cümleleri orijinal metinden seçip sıralar.

LexRank ve TextRank ise grafik tabanlı özetleme algoritmaları olup, cümleler arasındaki benzerlikleri temel alır. LexRank, cümleleri bir grafın düğümleri olarak modelleyen ve TF-IDF (Term Frequency-Inverse Document Frequency) yaklaşımıyla cümle benzerliklerini hesaplayan bir yöntemdir (Erkan ve Radev, 2004). İki cümle arasındaki benzerlik, bu cümlelerin graf üzerindeki bağlantı ağırlığını belirler. PageRank algoritmasına benzer şekilde, her düğüme (cümleye) diğer düğümlerden gelen bağlantıların sayısı ve ağırlığına göre bir önem skoru atanır. Bu skor, özete dahil edilecek cümlelerin seçiminde belirleyici olur. LexRank, TF-IDF ile az tekrar eden ancak bilgilendirici içeriğe sahip cümlelere öncelik vererek özlü özetler üretme potansiyeli taşır.

TextRank da benzer bir grafik temelli yaklaşım izlese de, cümle benzerliğini hesaplarken kelime örtüşmesinin yanı sıra sözcüksel ve anlamsal ilişkileri de dikkate alır (Mihalcea ve Tarau, 2004). Örneğin, aynı kavramsal alana ait eş anlamlı kelimeler veya ilişkili terimler içeren cümleler arasında bağlantılar kurabilir. Bu özelliğiyle LexRank'tan ayrışan TextRank, metnin ana fikirlerini daha az cümleyle özetleyebilir ve anlamsal derinliği yüksek çıktılar üretebilir.

Özetleme işlemine başlamadan önce, her üç algoritma için de metin, daha anlaşılır ve özetleme için uygun hale getirmek amacıyla bir dizi ön işleme adımından geçirilir. İlk olarak, makale metni parantez

içerikleri ve gereksiz karakterlerden arındırılır. Ardından, metin küçük harfe dönüştürülerek büyük-küçük harf ayrımından kaynaklanabilecek tutarsızlıklar giderilir. Bu ön işlemler, özetleme algoritmalarının metnin anlamını daha doğru bir şekilde kavramasına yardımcı olur. Ön işleme adımlarının ardından, temizlenmiş metin seçilen özetleme algoritmasına giriş olarak verilir. Her bir algoritma, kendi yöntemlerini kullanarak metni analiz eder, önemli cümleleri belirler ve metnin özlü bir özetini oluşturur. Bilimsel makalelerin özetlenmesinde, atıf bağlamı ve bilimsel söylem yapısının dikkate alınması, özetlerin daha anlamlı ve içerik açısından zengin olmasını sağlar (Cohan ve Goharian, 2018). Son olarak, oluşturulan özet, ilgili makalenin adıyla bir metin belgesi dosyasına kaydedilir. Kullanıcılara üç farklı özetleme algoritması seçeneği sunarak, farklı algoritmaların performansını kendi veri kümeleri ve ihtiyaçları doğrultusunda değerlendirmelerine olanak sağlanmıştır.

2.5. Metrikler ve Değerlendirme

Bu çalışmada, makale özetleme sistemlerinin performansını değerlendirmek için, özetlenen metinlerin orijinal metinlere ne kadar yakın olduğunu ölçen metrikler kullanılmıştır. Bu metrikler, özetleme sistemlerinin kalitesini objektif bir şekilde değerlendirilmesine olanak tanır. Bu bölümde, metin özetleme performansının değerlendirmesi için yaygın olarak kullanılan Meteor, Rouge ve Bleu hata metrikleri açıklanacaktır (Fabbri ve ark., 2021).

Metric for Evaluation of Translation with Explicit Ordering (Meteor): Özetlenen metin ile referans metin arasındaki kelime seçimi, sıralaması ve eşleşmesini dikkate alır. Metrik, kelimelerin kesin eşleşme (exact match), kök eşleşme (stem match) ve sözcük eşleşme (synonym match) gibi farklı eşleşme türlerine dayalı olarak hesaplanır. Sonuç olarak 0 ile 1 arasında bir değer alan bir Meteor skoru elde edilir. Yüksek skorlar, özetin orijinal metne daha yakın olduğunu gösterir (Banerjee ve Lavie, 2005).

Meteor, öncelikle özetlenen metin ile referans metin arasında mümkün olan tüm eşleşmeleri bulur. Bu eşleşmeler, farklı eşleşme türleri (kesin eşleşme, kök eşleşme, sözcük eşleşme) kullanılarak belirlenir. Daha sonra, her eşleşme türü için bir ağırlık faktörü hesaplanır. Meteor skoru Denklem 2'deki formül kullanılarak hesaplanır.

$$METEOR = (P \times R) / (\alpha \times P + (1 - \alpha) \times R) \quad (2)$$

Denklem 2'de P (precision): özetlenen metinde bulunan ve referans metinde bulunan eşleşen kelimelerin sayısı. R (Recall): referans metinde bulunan ve özetlenen metinde bulunan eşleşen kelimelerin sayısı. α : bir ağırlık faktörünü ifade eder ve genellikle 0,5 olarak seçilir.

Recall-Oriented Understudy for Gisting Evaluation (Rouge), metin özetleme sistemlerinin performansını değerlendirmek için yaygın olarak kullanılan bir değerlendirme aracıdır (Lin, 2004). Rouge, özet metin ile referans metin arasındaki benzerlikleri analiz eder ve çeşitli metrikler kullanarak

özetin bilgi içeriğini ve doğruluğunu ölçer. bu metrikler, özetlerin kalitesini değerlendirmede kritik bir öneme sahiptir.

Rouge-N metriği, özet metin ile referans metin arasındaki aynı N kelimelik dizilerin sayısını ölçer. N herhangi bir pozitif tam sayı olabilir ve genellikle unigram (Rouge-1), bigram (Rouge-2) gibi terimlerle ifade edilir. Rouge-N hesaplaması, özet metin ile referans metinde ortak olan N-gramların sayısının, referans metindeki toplam N-gram sayısına oranlanmasıyla yapılır. Rouge-N skorunun formülü Denklem 3'teki formül kullanılarak hesaplanır.

$$ROUGE - N = \frac{(\sum i=1^{n} Count(Unigram(S), Unigram(R)))}{n} \quad (3)$$

Denklem 3'te Count(Unigram(S), Unigram(R)): özetlenen metin (S) ve referans metin (R) içinde bulunan aynı N kelimelik dizilerin sayısı. n: referans metinde bulunan N kelimelik dizilerin sayısını ifade eder.

Rouge-L metriği ise, özetlenen metin ile referans metin arasındaki en uzun ortak alt dizinin (LCS - Longest Common Subsequence) uzunluğunu ölçer. LCS, özet ve referans metinlerdeki kelimelerin sırasını dikkate alarak, iki metin arasındaki en uzun ortak alt diziye karşılık gelir. Rouge-L, sıralı bilgiyi değerlendirir ve Denklem 4'teki formül kullanılarak hesaplanır.

$$ROUGE - L = \frac{2 * LCS(S,R)}{|S| + |R|} \quad (4)$$

Denklem 4'te $LCS(S, R)$: Özetlenen metin (S) ve referans metin (R) arasındaki en uzun ortak dizinin uzunluğu. $|S|$: Özetlenen metindeki kelime sayısı. $|R|$: Referans metindeki kelime sayısını ifade eder. Rouge metrikleri genellikle F1 skoruna dayalı olarak hesaplanır. F1 skoru, hatırlama (Recall) ve doğruluk (Precision) arasında bir denge sağlar. Rouge skorları, özet metnin referans metinle ne kadar iyi eşleştiğini gösterir ve 0 ile 1 arasında değer alır; 1 tam uyumu, 0 ise hiç uymadığını gösterir. Yüksek Rouge skorları, özetin referans metnin içeriğini doğru ve kapsamlı bir şekilde yansıttığını gösterir.

Bilingual Evaluation Understudy (Bleu) metriği, özetlenen metinlerin çevirisi olarak değerlendirilmesinde kullanılan bir ölçümdür. Bu metrik, referans metin ile özetlenen metin arasındaki kelime eşleşmelerini analiz ederek hesaplanır. Bleu, özellikle çeviri kalitesini değerlendirmede etkilidir ve referans metinde bulunan kelimelerin özetlenen metinde ne kadar doğru bir şekilde kullanıldığını ölçer. Bleu skoru 0 ile 1 arasında değer alır; yüksek skorlar, özetin referans metne daha yakın olduğunu gösterir (Papineni ve ark., 2002). Bleu, öncelikle, özetlenen metindeki kelimelerin, referans metindeki hangi kelimelerle eşleştiği belirlenir. Daha sonra, bu eşleşmelere dayalı olarak bir kelime eşleşme oranı hesaplanır. Bleu skoru, Denklem 5'teki formül kullanılarak hesaplanır:

$$BLEU = e^{(\sum i=1^{n} w_i * log(p_i))} \quad (5)$$

Denklem 5'te w_i: i-nci n-gram için bir ağırlık faktörü. p_i: i-nci n-gram için kelime eşleşme oranını ifade eder.

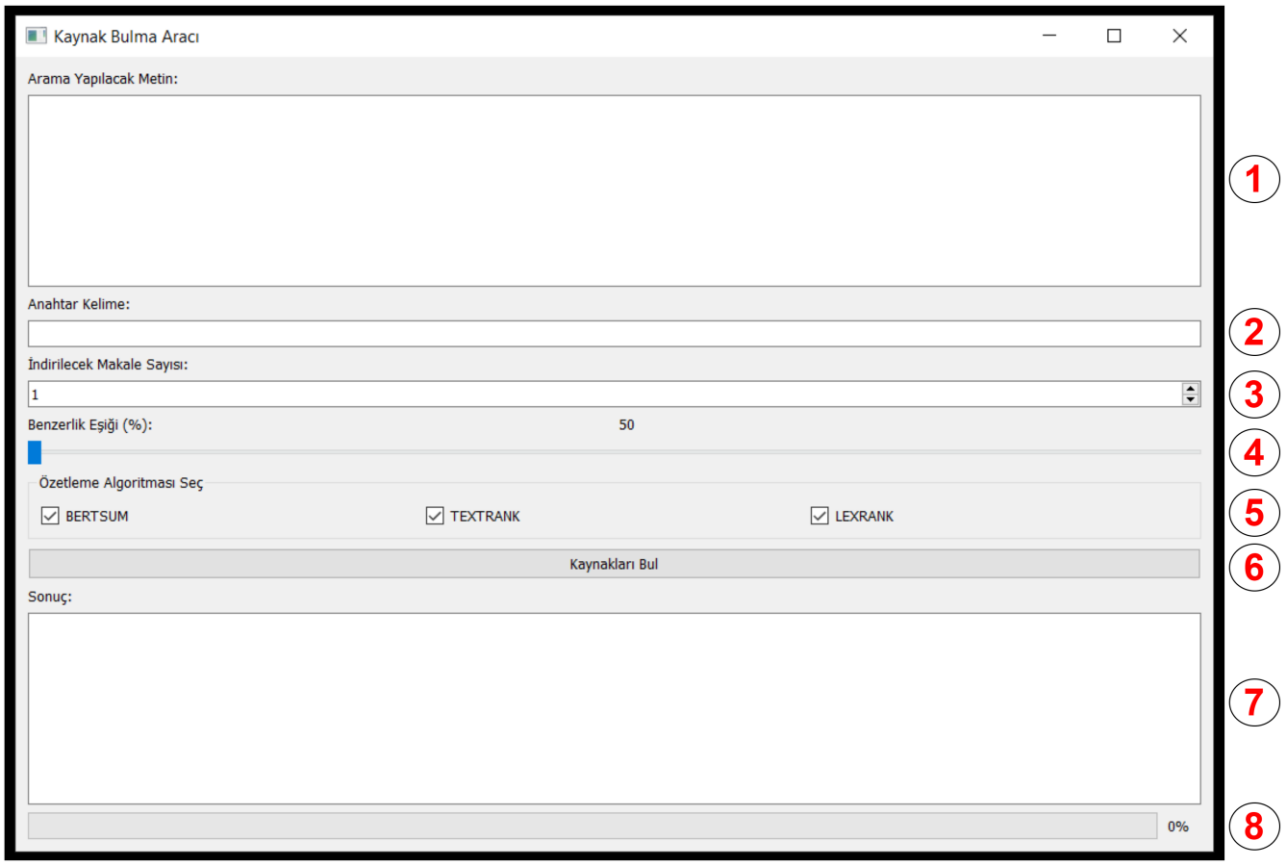
Bleu formülünde, farklı n-gramlar için farklı ağırlık faktörleri kullanılır. Örneğin, unigram (tekli kelimeler) için daha yüksek bir ağırlık faktörü kullanılırken, bigram (iki kelimelik diziler) için daha düşük bir ağırlık faktörü kullanılır. Bu metriklerin kullanılması, özetlenen metinlerin kalitesini objektif bir şekilde değerlendirmek için önemlidir. Değerlendirme sürecinde, PDF dosyalarından çıkarılan metinler kullanılarak oluşturulan özetler, makalelerin orijinal metinleriyle karşılaştırılmış ve Meteor, Rouge ve Bleu hata metrikleri hesaplanarak özetleme algoritmalarının performansı objektif bir şekilde değerlendirilmiştir.

3. Bulgular

Bu bölümde, geliştirilen prototip çalışmanın kullanıcı arayüzü, işleyişi ve çıktıları detaylı bir şekilde açıklanmıştır. Programın temel amacı, kullanıcıdan alınan referans metin ve anahtar kelimeye bağlı olarak, Google Scholar platformundan makaleleri incelemek, özetlemek ve bu özetlerin referans metinle benzerliklerini hesaplamaktır.

3.1. Kullanıcı Arayüzü ve İşlevselliği

Programın kullanıcı arayüzü, PyQt5 kütüphanesi kullanılarak geliştirilmiş olup kullanıcı dostu ve işlevsel bir yapıya sahiptir. Arayüz, kullanıcıların gerekli bilgileri kolayca girebilmelerini ve programın sunduğu farklı seçenekler arasında seçim yapabilmelerini sağlayarak programın kullanımını kolaylaştırır. Şekil 2'de gösterilen arayüz, kullanıcıların metin benzerliği analizi ve metin özetleme işlemlerini verimli bir şekilde gerçekleştirebilmelerini sağlayan bir dizi bileşenden oluşmaktadır.



Şekil 2. Çalışmanın Kullanıcı Arayüz Tasarımı

Şekil 2'de görüldüğü üzere, arayüzün üst kısmında (1) kullanıcı tarafından arama yapılacak metnin girileceği geniş bir metin kutusu yer almaktadır. Bu metin kutusu, benzerlik analizi için kullanılacak temel metni temsil eder. Arayüzün ortasında ise kullanıcıdan alınacak diğer girdiler için üç ayrı alan bulunmaktadır: (2) Anahtar Kelime, (3) İndirilecek Makale Sayısı ve (4) Benzerlik Eşiği.

(2) Anahtar Kelime alanına, Google Scholar platformunda arama yapmak için kullanılacak anahtar kelimeler girilir. (3) İndirilecek Makale Sayısı alanında, Google Scholar aramasından kaç adet makalenin indirileceği belirtilir. (4) Benzerlik Eşiği (%) alanında ise, benzerlik analizi için kullanılacak eşik değeri yüzdelik olarak (50-100 arası) belirlenir. Bu eşik değeri, referans metin ile özetlenen metinler arasındaki benzerlik skorlarının karşılaştırılacağı bir sınır değerini temsil eder.

Arayüzün alt kısmında (5) Özetleme Algoritması Seç başlıklı bir bölüm yer almaktadır. Bu bölümde, kullanıcılar metin özetleme işlemi için kullanmak istedikleri algoritmayı seçebilirler. Bertsum, TextRank ve LexRank olmak üzere üç farklı özetleme algoritması seçeneği sunulmaktadır. Kullanıcılar, bu algoritmalardan birini veya daha fazlasını seçerek özetleme işlemini gerçekleştirebilirler. Arayüzün en alt kısmında ise (6) Kaynakları Bul butonu, (7) Sonuç bölümü ve (8) ilerleme çubuğu yer almaktadır. Kullanıcılar, tüm girdileri doldurduktan ve özetleme algoritması seçimini yaptıktan sonra (6) Kaynakları Bul butonuna tıklayarak programı çalıştırırlar. Programın çalışması sırasında, indirme ve işleme süreçlerini gösteren bir (8) ilerleme çubuğu görüntülenir. Benzerlik analizi sonuçları (7) Sonuç bölümünde görüntülenir ve ayrıca bir Excel dosyasına kaydedilir.

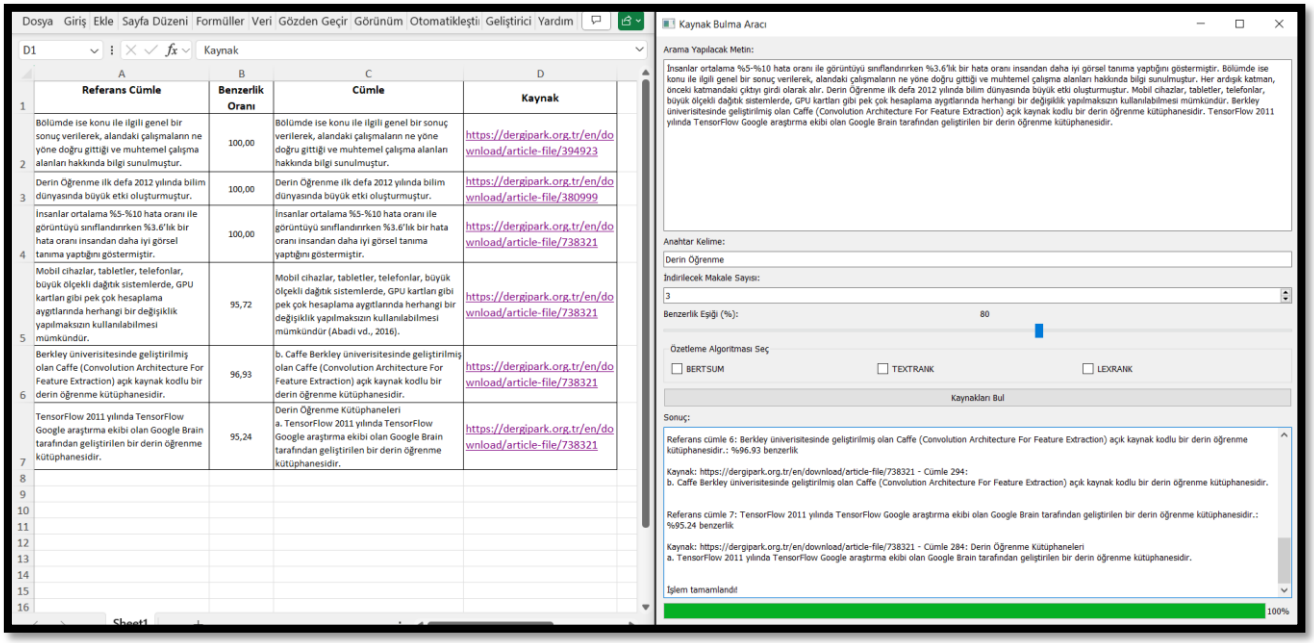
3.2. Program İşleyişi

Program, kullanıcı dostu arayüzü aracılığıyla alınan girdilere göre çalışır ve otomatik olarak metin benzerliği analizi ve özetleme işlemlerini gerçekleştirir. İlk olarak, kullanıcı tarafından belirtilen anahtar kelimeler ve makale sayısı kullanılarak, Selenium kütüphanesi aracılığıyla Google Scholar platformunda otomatik bir arama gerçekleştirilir ve belirtilen sayıda makaleye ait PDF linkleri toplanır. Ardından, Requests kütüphanesi kullanılarak bu makaleler indirilir ve yerel diske kaydedilir. İndirilen PDF dosyaları, PDFPlumber kütüphanesi kullanılarak işlenir ve metin içerikleri çıkarılır. Bu işlem sırasında, *Kaynakların İndirilmesi ve İşlenmesi* 'nde detaylı olarak açıklanan ön işleme adımları uygulanır. Parantez içerikleri, tablo ve şekil başlıkları gibi gereksiz bilgiler metinden temizlenir. Metin küçük harfe dönüştürülür, sayısal değerler çıkarılır ve metnin dili otomatik olarak tespit edilir. Dil tespitinin ardından, NLTK kütüphanesi kullanılarak metin tokenlere ayrılır ve dile uygun durak kelimeler (stopwords) metinden çıkarılır.

Ön işleme adımlarından geçen temizlenmiş metinler, kullanıcı tarafından seçilen özetleme algoritması veya algoritmaları kullanılarak özetlenir. Program, Bertsum, LexRank ve TextRank olmak üzere üç farklı özetleme algoritması seçeneği sunar ve kullanıcı, bu algoritmalardan birini veya daha fazlasını seçebilir. Her bir algoritma, kendi yöntemlerini kullanarak metni analiz eder, önemli cümleleri belirler ve metnin özlü bir özetini oluşturur. Oluşturulan özetler, ilgili makalenin adıyla birlikte ayrı metin dosyalarına kaydedilir. Referans metin ve özetlenen metinler arasındaki benzerlik skorları, TF-IDF vektörleri ve kosinüs benzerliği kullanılarak hesaplanır. Referans metin ve her bir özetlenen metin cümlelere ayrılır ve TfidfVectorizer sınıfı kullanılarak her bir cümle için TF-IDF vektörleri oluşturulur. Ardından, referans metindeki her cümle ile özetlenen metinlerdeki her cümle arasında kosinüs benzerliği hesaplanır. Hesaplanan kosinüs benzerlik skorları, kullanıcı tarafından belirlenen benzerlik eşiği ile karşılaştırılır. Eşik değerini aşan skorlara sahip cümleler, referans metin ile anlamlı bir benzerlik taşıdığı kabul edilir ve programın arayüzündeki Sonuç bölümünde detaylı bir şekilde sunulur. Ayrıca, benzerlik analizi sonuçları ve özetlenen metinler bir Excel dosyasına kaydedilir, böylece kullanıcılar daha sonra bu bilgilere kolayca erişebilirler.

3.3. Benzerlik Analizi

Program; belirtilen anahtar kelimelerle ilgili makaleleri Google Scholar'dan indirip özetledikten sonra, referans metin ile özetlenen metinler arasında cümle düzeyinde benzerlik analizi gerçekleştirir. Bu analiz, kullanıcı tarafından belirlenen benzerlik eşiğine göre yapılır. Eşik değeri aşan benzerlik skorlarına sahip cümleler, referans metinle anlamlı bir benzerlik taşıdığı kabul edilir ve programın arayüzündeki Sonuç bölümünde ve sonuc.xlsx adlı bir Excel dosyasında detaylı bir şekilde sunulur. Kullanıcılar, metin özetleme işlemini gerçekleştirmeden de doğrudan indirilen makalelerin tam metinleri üzerinde benzerlik analizi çalıştırabilirler. Şekil 3'te, programın arayüzü ve sonuc.xlsx dosyasından alınan ekran görüntüleri gösterilmektedir.



Şekil 3. Tespit edilen cümle benzerliğinin kaynak ile birlikte excel'de ve kullanıcı arayüzünde gösterilmesi

Kullanıcı arayüzünde, "Arama Yapılacak Metin" bölümüne, intihal kontrolü yapılacak referans metin girilmiştir. Örnek olarak, "Derin Öğrenme" anahtar kelimesi ile ilgili 3 makale indirilmiş, özetleme algoritması seçilmemiş ve benzerlik eşiği %80 olarak belirlenmiştir. Benzerlik analizi sonucunda, referans metindeki 6. ve 7. cümleler ile indirilen makalelerden birinin özetindeki cümleler arasında yüksek benzerlik tespit edilmiştir. Program arayüzünde, her bir eşleşme için referans cümle numarası, benzerlik oranı, kaynak makalenin linki ve cümle numarası görüntülenir. Örneğin, "Berkeley üniversitesinde geliştirilmiş olan Caffe (Convolution Architecture For Feature Extraction) açık kaynak kodlu bir derin öğrenme kütüphanesidir." cümlesi ile indirilen makalelerden birindeki 294. cümle olan "b. Caffe Berkeley üniversitesinde geliştirilmiş olan Caffe (Convolution Architecture For Feature Extraction) açık kaynak kodlu bir derin öğrenme kütüphanesidir." cümlesi arasında %96,93 benzerlik tespit edilmiştir. Bu eşleşme, referans metindeki cümlenin, indirilen makaledeki bir cümle ile neredeyse birebir aynı olduğunu gösterir. sonuc.xlsx dosyasında ise benzerlik analizi sonuçları tablo formatında sunulur. Bu tabloda, her bir referans cümle için, hedef makaledeki en yüksek benzerlik skoruna sahip cümle ve bu skorun oranı gösterilmektedir. Her bir satır bir benzerlik eşleşmesini temsil eder ve şu bilgileri içerir: Referans Cümle, Benzerlik Oranı, Cümle ve Kaynak. Bu şekilde, kullanıcılar hem program arayüzünde hem de Excel dosyasında detaylı benzerlik analizi sonuçlarını inceleyebilir, olası intihal durumlarını tespit edebilir ve kaynakları doğru bir şekilde belirleyebilirler.

3.4. Özetlerin Değerlendirilmesi

Metin özetleme, büyük miktarda yazılı içeriğin ana fikirlerini kısa ve öz bir şekilde aktarma sürecidir. Bu çalışma, metin özetleme algoritmalarının doğruluğunu çeşitli değerlendirme metrikleri ile analiz etmektedir. Bertsum, TextRank ve LexRank algoritmaları, farklı bilim alanlarını ve dili temsilen seçilen

dört alt başlık (Lineer Cebir, Kültürel Miras ve Görüntü İşleme, Deep Learning Architectures) ve 12 farklı makale üzerinden incelenmiştir. Bu alanlar, sırasıyla sayısal, sözel ve yazılım ana başlıklarını ve Türkçe ile İngilizce dillerini temsil etmektedir. Algoritmaların özetleme performansları, farklı makaleler kullanılarak Meteor, Bleu, Rouge-1, Rouge-2 ve Rouge-L metrikleri ile karşılaştırılmıştır. Elde edilen skorlar, Tablo 1'de sunulmaktadır. Bu analiz, özetleme algoritmalarının performansını değerlendirmek ve farklı metriklerin bu performans üzerindeki etkisini anlamak açısından önem taşımaktadır.

Tablo 1. Metin özetleme kapsamında oluşturulan doğruluk sonuçları

Anahtar Kelime	Makale Kodu	Özetleme Algoritması	Meteor	Bleu	Rouge-1	Rouge-2	Rouge-L
Lineer Cebir	1	Bertsum	0,678	0,443	0,779	0,691	0,463
		TextRank	**0,714**	**0,484**	**0,822**	**0,731**	**0,503**
		LexRank	0,584	0,473	**0,822**	0,720	0,172
	2	Bertsum	0,721	0,540	0,890	0,842	0,873
		TextRank	**0,804**	**0,625**	**0,929**	**0,887**	**0,909**
		LexRank	0,566	0,597	**0,929**	0,843	0,212
	3	Bertsum	0,842	0,614	0,849	0,804	0,832
		TextRank	**0,844**	**0,628**	**0,854**	**0,811**	**0,839**
		LexRank	0,664	0,606	**0,854**	0,788	0,216
Kültürel Miras	4	Bertsum	0,733	0,592	0,785	0,764	0,777
		TextRank	**0,757**	**0,655**	**0,834**	**0,814**	**0,826**
		LexRank	0,659	0,641	**0,834**	0,800	0,267
	5	Bertsum	**0,824**	0,827	0,921	0,911	0,910
		TextRank	0,823	**0,831**	**0,923**	**0,913**	**0,911**
		LexRank	0,674	0,810	**0,923**	0,897	0,211
	6	Bertsum	0,617	0,448	0,917	0,909	0,916
		TextRank	**0,783**	**0,509**	**0,966**	**0,961**	**0,965**
		LexRank	0,507	0,493	**0,966**	0,939	0,215
Görüntü İşleme	7	Bertsum	0,812	0,637	0,859	0,835	0,850
		TextRank	**0,818**	**0,699**	**0,885**	**0,861**	**0,866**
		LexRank	0,620	0,663	**0,885**	0,821	0,214
	8	Bertsum	0,885	0,637	0,826	0,800	0,800
		TextRank	**0,913**	**0,656**	**0,834**	**0,810**	**0,808**
		LexRank	0,688	0,629	**0,834**	0,783	0,199
	9	Bertsum	0,683	0,516	0,750	0,724	0,738
		TextRank	**0,694**	**0,555**	**0,774**	**0,748**	**0,754**
		LexRank	0,582	0,534	**0,774**	0,724	0,187
Deep Learning Architectures	10	Bertsum	0,874	0,880	0,925	0,904	0,918
		TextRank	**0,913**	**0,905**	**0,940**	**0,914**	**0,933**
		LexRank	0,589	0,880	**0,940**	0,881	0,247
	11	Bertsum	0,820	0,613	0,897	0,774	0,733
		TextRank	**0,854**	**0,685**	**0,945**	**0,811**	**0,760**
		LexRank	0,662	0,662	**0,945**	0,747	0,222
	12	Bertsum	0,622	0,655	0,890	0,784	0,461
		TextRank	**0,631**	**0,732**	**0,930**	**0,812**	**0,465**
		LexRank	0,535	0,703	**0,930**	0,776	0,175

Çalışmada, "Lineer Cebir", "Kültürel Miras", "Görüntü İşleme" ve “Deep Learning Architectures” başlığı altında toplam 12 makale incelenmiştir. Her makale için Bertsum, TextRank ve LexRank algoritmaları kullanılarak özetler oluşturulmuş ve bu özetlerin kalitesi beş farklı metrik ile değerlendirilmiştir.

3.4.1. Doğruluk Metriklerinin Değerlendirilmesi

Genel olarak, Bertsum ve TextRank algoritmalarının LexRank'a kıyasla daha yüksek sonuçlar gözlemlenmiştir. Özellikle Rouge-L metriğinde LexRank'ın diğer iki metriğe göre belirgin şekilde düşük skorlar almıştır. Örneğin, 2 numaralı makalede Bertsum ve TextRank sırasıyla 0,873 ve 0,909 Rouge-L skorları alırken, LexRank yalnızca 0,212 almıştır. Bu fark, LexRank'ın uzun bağlantılı cümle yapılarını yakalamada diğer iki algoritmaya göre daha az başarılı olduğunu göstermektedir.

Rouge-1 metriği genellikle en yüksek skorları verirken, Bleu metriği çoğunlukla en düşük skorları vermiştir. Örneğin, 5 numaralı makalede Bertsum algoritması için Rouge-1 skoru 0,921 iken, Bleu skoru sadece 0,827'dir. Bu belirgin fark, Rouge-1'in tek kelime eşleşmelerine odaklanırken, Bleu'nun daha uzun n-gram eşleşmelerini dikkate almasından kaynaklanıyor olabilir. Bu durum, metriklerin farklı özelliklere odaklandığını ve tek bir metriğe bağlı kalmanın yanıltıcı olabileceğini göstermektedir. Dolayısıyla, özetleme performansını değerlendirirken birden fazla metriği birlikte kullanmak daha sağlıklı sonuçlar verecektir.

3.4.2. Konu Bazlı Analiz

"Kültürel Miras" konulu makalelerde (4, 5, 6 numaralı) tüm metriklerin ortalamaları alındığında diğer konulara göre daha yüksek skorlar elde ettiği görülmüştür. Örneğin, 6 numaralı makalede TextRank algoritması Rouge-1'de 0,966, Rouge-2'de 0,961 ve Rouge-L'de 0,965 gibi oldukça yüksek skorlar elde etmiştir. Bu durum, kültürel miras metinlerinin yapısının özetleme algoritmalarına daha uygun olabileceğini düşündürmektedir. Sözel metinlerinin muhtemelen daha betimleyici ve tekrar eden ifadeler içermesi, özetleme algoritmalarının kilit noktaları daha kolay yakalamasını sağlıyor olabilir.

3.4.3. Algoritma Tutarlılığı

Algoritmaların performansı makaleler arasında değişkenlik göstermekle birlikte, bazı tutarlı örüntüler gözlemlenmiştir. LexRank algoritması, özellikle Rouge-L metriğinde sürekli olarak en düşük skorları almıştır. Bu durum, LexRank'ın uzun ve bağlantılı cümle yapılarını yakalamada diğer iki algoritmaya göre daha zayıf olduğunu göstermektedir.

TextRank algoritması, birçok durumda en yüksek skoru elde etmiştir. Özellikle Rouge-1 ve Rouge-2 metriklerinde tutarlı bir şekilde yüksek performans göstermiştir. Örneğin, 2 numaralı makalede TextRank, Rouge-1'de 0,929 ve Rouge-2'de 0,887 gibi yüksek skorlar elde etmiştir. Bu tutarlılık, TextRank'ın farklı konu alanlarında ve metriklerde genel olarak iyi performans gösterdiğini ortaya koymaktadır. Bertsum algoritması ise genel olarak iyi performans göstermekle birlikte, makaleler arasında daha fazla değişkenlik göstermiştir. Bazı makalelerde (örneğin 5 numaralı makale) en yüksek Meteor skorunu elde ederken, diğerlerinde daha düşük skorlar almıştır. Bu durum, Bertsum'un performansının metin içeriğine ve yapısına daha duyarlı olabileceğini göstermektedir.

"Deep Learning Architectures" kategorisinde, Bertsum ve TextRank algoritmalarının 10 numaralı makalede benzer performans sergilediği gözlemlenmektedir. Bertsum, 0,874 Meteor skoru ile dikkat

çekerken, TextRank 0,913 Meteor skoru ile bu algoritmayı geçmiştir. Ancak, 11 ve 12 numaralı makalelerde, TextRank yine Rouge-1 ve Rouge-2 metriklerinde en yüksek puanları alarak öne çıkmıştır. Özellikle, 11 numaralı makalede TextRank 0,945 Rouge-1 ve 0,811 Rouge-2 skoru ile Bertsum'un önüne geçmiştir. Bu, TextRank'ın farklı makalelerde daha tutarlı ve güçlü performans gösterdiğinin bir başka örneğidir. Ayrıca, 12 numaralı makalede de TextRank'ın Rouge-1 ve Rouge-2 metriklerinde Bertsum'u geride bıraktığı ve LexRank'ın bu makalede diğer algoritmaların çok gerisinde kaldığı görülmektedir. Bu noktada, çalışma dili ve metinlerin yapısı üzerine yapılan analizler, algoritmaların İngilizce özetleme görevlerinde de başarılı sonuçlar elde edebildiğini göstermektedir. Özellikle "Deep Learning Architectures" alanında incelenen makaleler, İngilizce metinler içermektedir ve algoritmaların farklı dillerdeki performansları kıyaslandığında, İngilizce metinlerde de benzer başarı oranları yakalandığı gözlemlenmiştir. Makale dili otomatik olarak Python programlama dilindeki "langdetect" kütüphanesi ile tespit edilmiş, bu sayede özetleme algoritmaları, dil bağımsız olarak doğru şekilde yönlendirilmiştir. İngilizce metin özetleme görevlerinde Bertsum ve TextRank algoritmaları, Rouge ve Meteor gibi metriklerde yüksek puanlar alarak etkili performans sergilemektedir. Örneğin, İngilizce dilinde oluşturulan özetlerde, metin yapısının daha düzenli ve n-gram tabanlı metriklerin bu yapıyı daha iyi yakalayabilmesi nedeniyle TextRank algoritmasının Rouge-1 ve Rouge-2 metriklerinde öne çıktığı görülmektedir. Bertsum ise, hem İngilizce hem de Türkçe metinlerde genel olarak güçlü bir performans sergilemekte, ancak metinlerin içerik yoğunluğu ve yapısal farklılıklarına daha duyarlı olduğu için bazı İngilizce metinlerde daha değişken sonuçlar verebilmektedir. İngilizce metinlerde elde edilen bu başarı, algoritmaların dil bağımsız bir şekilde özetleme yapabildiği ve özellikle büyük dil modellerinin dil yapısını anlamada etkin olduğu gerçeğini desteklemektedir. Bu durum, gelecekte yapılacak çalışmalarda metin özetleme algoritmalarının dil yetkinliklerinin artırılmasının önemini vurgulamaktadır.

Bu analiz, özetleme algoritmalarının performansının konu alanına ve kullanılan değerlendirme metriğine göre değişkenlik gösterdiğini ortaya koymaktadır. Bertsum ve TextRank algoritmaları genel olarak daha iyi performans sergilese de, LexRank'ın da belirli durumlarda etkili olabildiği görülmüştür. Gelecekteki çalışmalarda, farklı konu alanlarına özgü özetleme algoritmalarının geliştirilmesi ve metrik seçiminin özetleme performansına etkisinin daha detaylı incelenmesi önerilmektedir.

4. Sonuçlar

Bu çalışmada; akademik araştırma süreçlerinde bilgi edinme ve kaynak bulma verimliliğini artırmayı amaçlayan, Google Scholar platformu üzerinden çalışan ve BERT tabanlı metin özetleme ile benzerlik analizi tekniklerini entegre eden özgün bir prototip geliştirilmiştir. Makine öğrenmesi ve derin öğrenme tabanlı modellerin, literatür taramaları ve metin özetleme süreçlerinde kullanımı, araştırmacılara daha hızlı, verimli ve kapsamlı analizler sunma imkânı sağlamaktadır (Gulati ve ark., 2023; Edrees ve Ortakci, 2024).

Geliştirilen prototip, kullanıcı dostu bir arayüz sunan PyQt5 kütüphanesi ile tasarlanmıştır. Bu arayüz, kullanıcıların araştırma süreçlerini kolaylaştırmak amacıyla Bertsum, LexRank ve TextRank gibi üç

farklı özetleme algoritması arasından seçim yapma olanağı sunmaktadır. Kullanıcılar, arayüz aracılığıyla araştıracakları metinleri ve anahtar kelimeleri sisteme girerek, Google Scholar üzerinden ilgili makaleleri indirebilir, bu makalelerin özetlerini oluşturabilir ve girdikleri metinlerle benzerlik analizlerini gerçekleştirebilirler. Bu çalışmanın en önemli katkısı, metin özetleme ve benzerlik analizini tek bir sistemde bir araya getiren entegre bir çözüm sunmasıdır. Literatürde, metin özetleme ve benzerlik analizi genellikle ayrı ayrı ele alınırken, bu çalışma her iki işlevi birleştirerek, araştırmacılara daha kapsamlı ve verimli bir araştırma deneyimi sunmayı hedeflemektedir. Ayrıca, kullanıcıların özetleme algoritmaları arasında seçim yapabilmesi ve benzerlik analizindeki eşiği belirleyebilmesi uygulamanın esnekliğini artıran önemli bir özellik olarak öne çıkmaktadır. Özellikle, prototipin sunduğu özetleme ve benzerlik analizlerinin doğruluk sonuçları, uygulamanın yüksek kaliteli özetler ürettiğini ve referans metinlerle anlamlı benzerlikler taşıyan cümleleri başarılı bir şekilde tespit edebildiğini göstermektedir. Ayrıca, makale tespiti için kullanılan Python kütüphanesi "langdetect" sayesinde, çalışma dili otomatik olarak algılanarak İngilizce metin özetlemelerinde de başarılı sonuçlar elde edilmiştir. Bu sayede, uygulama hem Türkçe hem de İngilizce gibi farklı dillerdeki metinlerde etkin bir şekilde çalışabilmektedir.

Uygulamanın akademik araştırma süreçlerinde kullanımını kolaylaştırmak ve hızlandırmak için geliştirilmiş olmasıyla birlikte, etik kullanımına da dikkat çekilmesi gerekmektedir. Benzerlik analizleri, olası intihal tespitinde kullanılabilir, ancak bu sonuçların tek başına yeterli olmadığının altı çizilmelidir. İntihal değerlendirmeleri yapılırken, bağlam, niyet ve diğer kanıtların da dikkate alınması büyük önem taşır. Bu uygulamanın temel amacı, literatür tarama süreçlerini daha verimli hale getirerek araştırmacılara yardımcı olmak ve kaynak bulma sürelerini optimize etmektir.

Geliştirilen bu prototip çalışma genel bir akademik makale veri kümesi üzerinde test edilmiştir. Ancak, gelecekte daha geniş çaplı ve alana özgü veri kümeleri üzerinde uygulamanın performansı artırılabilir. Metin çıkarma süreçlerinde PDF dosyalarının yapısal farklılıklarından kaynaklanan hatalar oluşabilmekte ve bu durum özetleme ile benzerlik analizlerinin doğruluğunu etkileyebilmektedir. Bu sorunun çözülmesi için daha gelişmiş metin çıkarma algoritmalarının entegrasyonu ve farklı PDF formatlarının daha etkin işlenmesi gerekmektedir.

Gelecekte uygulamanın işlevselliğini artırmak adına farklı akademik dergi platformlarının entegrasyonu, benzer metinlerden otomatik referans ekleme ve daha fazla özetleme algoritmasının karşılaştırılması gibi yenilikler eklenebilir. Ayrıca, hata metriklerinin uygulamaya dahil edilmesi ve kullanıcıların algoritmaların performansını anahtar kelimeler ve metrikler bazında değerlendirmesine imkan sağlayan bir arayüz geliştirilmesi, uygulamanın kullanımını daha verimli hale getirecektir. Böylelikle, araştırmacılar ihtiyaçlarına en uygun özetleme algoritmasını bilinçli bir şekilde seçebilecektir. Bu gelişmeler uygulamanın akademik araştırma süreçlerine daha fazla entegre olmasını sağlayarak, araştırmacılar için daha kapsamlı ve faydalı bir araç haline gelmesine katkı sağlayacaktır. Tablo 2, bu çalışmanın literatürdeki diğer çalışmalarla karşılaştırmasını sunmaktadır. Tablo, prototipin özgün katkılarını ve benzer çalışmalardan farklı yönlerini göstermektedir.

Tablo 2. Metin özetleme kapsamında oluşturulan doğruluk sonuçları

Özellik	Literatürdeki Diğer Çalışmalar	Bu Çalışma (Prototip)
Temel Amaç	Edrees ve arkadaşları, Metin özetleme amacıyla cümle kümeleme ve doğal dil işleme tekniklerinden faydalanmıştır (Edrees ve ark., 2024).	Literatür tarama süreçlerini hızlandırmayı, metin analizi ve özetleme işlevlerini Google Scholar platformu ile entegre ederek kullanıcı dostu ve kapsamlı bir araç sunmayı hedeflemektedir.
Veri Kaynağı	Orel ve arkadaşları akademik araştırma süreçlerini hızlandırmak için otomatik özetleme ve literatür inceleme sistemi geliştirilmiştir (Orel ve ark., 2023). Gulati ve arkadaşları Medium bloglarından derlenen bir veri seti üzerinde özetleme gerçekleştirilmiştir. (Gulati ve ark., 2023)	Google Scholar platformu üzerinden, kullanıcının seçtiği konularla ilgili en güncel çalışmalara otomatik olarak erişmesini sağlayan bir yöntem önerilmektedir. Bu yaklaşım, akademik araştırma süreçlerinde daha kapsamlı ve güncel bir bilgi havuzu oluşturmayı hedeflemektedir.
Özetleme Algoritmaları	Aftiss ve arkadaşları PageRank algoritması ve Longformer modeli ile özetleme yapmıştır (Aftiss ve ark., 2024). Thirumahal TF-IDF tabanlı kümeleme yöntemleri kullanılarak özetleme gerçekleştirilmiştir; algoritma kıyaslamasına yer verilmemiştir (Thirumahal, 2024).	Bertsum, TextRank ve LexRank algoritmalarını karşılaştırmalı olarak sunarak, farklı özetleme yöntemlerinin performansını değerlendirme imkânı sağlamaktadır. Araştırmacıların veri kümelerine en uygun algoritmayı seçebilmesi için kullanıcı odaklı bir yaklaşım benimsemektedir.
Değerlendirme Metrikleri	Kahdim ve arkadaşları puanlama içeren bir özetleme modeli önermiştir ve ROUGE-1, ROUGE-2, ROUGE-L ile değerlendirme gerçekleştirmiştir (Kahdim ve ark., 2025).	Rouge, Bleu ve Meteor gibi metrikler aracılığıyla kapsamlı bir özet kalitesi değerlendirmesi yapılmaktadır. Bu metrikler, farklı algoritmaların performansını karşılaştırmayı ve sistem çıktılarını daha detaylı bir şekilde analiz etmeyi mümkün kılmaktadır.
Arayüz ve Kullanım	Naing ve arkadaşları Google Scholar'dan makale toplama süreci için Selenium tabanlı bir sistem geliştirilmiştir; ancak kapsamlı bir arayüz veya özetleme ve benzerlik analizi entegrasyonu bulunmamaktadır (Naing ve ark., 2023)	PyQt5 tabanlı bir arayüz sunarak kullanıcıların anahtar kelime girişi, metin özetleme algoritması seçimi ve benzerlik analizi işlemlerini tek bir platform üzerinden kolayca gerçekleştirmelerine olanak tanımaktadır. Bu kullanıcı dostu yapı, tüm süreçleri entegre bir şekilde yönetmeyi mümkün kılmaktadır.
Benzerlik Analizi	Sofi ve Selamat LDA ve TF-IDF kullanarak duygu analizi gerçekleştirmiştir. (Sofi ve Selamat, 2023) Setiawan ve Adnyana yardım masası sohbet robotlarında TF-IDF ve kosinüs benzerliği yöntemlerini uygulamıştır (Setiawan ve Adnyana, 2023).	Cümle düzeyinde TF-IDF ve kosinüs benzerliği yöntemlerini kullanarak, eşik değerine göre özelleştirilebilen bir benzerlik analizi sunmaktadır. Bu yapı, farklı araştırma ihtiyaçlarına uyarlanabilir bir çözüm sağlamaktadır.

Bu çalışmanın özgün katkısı, literatür taraması, PDF doküman işleme, metin benzerliği analizi, özetleme ve değerlendirme süreçlerini tek bir platformda entegre ederek, akademik araştırmalar için kapsamlı bir çözüm sunmasıdır. Google Scholar entegrasyonu, TF-IDF tabanlı benzerlik analizi ve Bertsum tabanlı

özetleme yöntemlerinin bir arada kullanılması, bu çalışmayı literatürdeki diğer çalışmalardan ayıran temel özelliklerdir.

Geliştirilen prototipin kaynak kodu GitHub platformu üzerinden açık kaynak olarak https://github.com/OnurSahinCeng/Kaynak-Bulma-Arac-Prototip bağlantısında paylaşılmıştır. Bu paylaşım, bilimsel çalışmaların tekrar edilebilirliğini artırmak ve prototipin daha geniş kitleler tarafından kullanılmasını sağlamak amacıyla yapılmıştır. İlerleyen süreçte, kullanıcı geri bildirimlerine dayalı olarak prototipin sürekli güncellenmesi ve daha fazla akademik platformla entegre edilmesi hedeflenmektedir.

Çıkar Çatışması Beyanı

Makale yazarları aralarında herhangi bir çıkar çatışması olmadığını beyan ederler.

Araştırmacıların Katkı Oranı Beyan Özeti

Bu çalışmanın araştırma, geliştirme ve kodlama süreçleri Yazar1 tarafından, araştırma ve düzenleme süreçleri Yazar2 tarafından yapılmıştır.

Kaynakça

Aftiss A., Lamsiyah S., El Alaoui SO., Schommer C. Biomdsum: an effective hybrid biomedical multi-document summarization method based on pagerank and longformer encoder-decoder. IEEE Access 2024; 12.

Alzahrani SM., Salim N., Abraham A. Understanding plagiarism linguistic patterns, textual features, and detection methods. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 2011; 42(2): 133-149.

Banerjee S., Lavie A. Meteor: an automatic metric for MT evaluation with improved correlation with human judgments. In: Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization 2005; 65-72.

Beel J., Gipp B., Langer S., Breitinger C. Paper recommender systems: a literature survey. International Journal on Digital Libraries 2016; 17: 305-338.

Bird S., Klein E., Loper E. Natural language processing with python: Analyzing text with the natural language toolkit. O'Reilly Media, Inc. 2009.

Boell SK., Cecez-Kecmanovic D. A hermeneutic approach for conducting literature reviews and literature searches. Communications of the Association for Information Systems 2014; 34(1): 12.

Cao Z., Li W., Wei F., Li S. Retrieve, rerank and rewrite: soft template based neural summarization. In: Proceedings of the Association for Computational Linguistics (ACL) 2018.

Cohan A., Dernoncourt F., Kim DS., Bui T., Kim S., Chang W., Goharian N. A discourse-aware attention model for abstractive summarization of long documents. arXiv preprint arXiv:1804.05685 2018.

Cohan A., Goharian N. Scientific document summarization via citation contextualization and scientific discourse. International Journal on Digital Libraries 2018; 19: 287-303.

Creswell JW., Creswell JD. Mixed methods procedures. In: Research Design: Qualitative, Quantitative, and Mixed Methods Approaches 2018.

Devlin J. Bert: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 2018.

Edress Z., Ortakci Y. Optimizing text summarization with sentence clustering and natural language processing. International Journal of Advanced Computer Science & Applications 2024; 15(10).

Erkan G., Radev DR. Lexrank: graph-based lexical centrality as salience in text summarization. Journal of Artificial Intelligence Research 2004; 22: 457-479.

Fabbri AR., Kryściński W., McCann B., Xiong C., Socher R., Radev D. Summeval: re-evaluating summarization evaluation. Transactions of the Association for Computational Linguistics 2021; 9: 391-409.

Fateh Ali N., Mohtasim M., Mosharrof S., Gopi Krishna T. Automated literature review using nlp techniques and llm-based retrieval-augmented generation. arXiv e-prints 2024; arXiv-2411.

Fauzi R., Iqbal M., Haryanti T. Design and implementation of a final project plagiarism detection system using cosine similarity method. International Journal of Applied Information Technology 2021; 59-74.

Ghanem FA., Padma MC., Abdulwahab HM., Alkhatib R. Novel genetic optimization techniques for accurate social media data summarization and classification using deep learning models. Technologies 2024; 12(10): 199.

Gulati V., Kumar D., Popescu DE., Hemanth JD. Extractive article summarization using integrated TextRank and BM25+ algorithm. Electronics 2023; 12(2): 372.

Gusenbauer M., Haddaway NR. Which academic search systems are suitable for systematic reviews or meta-analyses? evaluating retrieval qualities of Google Scholar, PubMed, and 26 other resources. Research Synthesis Methods 2020; 11(2): 181-217.

Hemamou L., Debiane M. Scaling up summarization: leveraging large language models for long text extractive summarization. arXiv preprint arXiv:2408.15801 2024.

Januzaj Y., Luma A. Cosine similarity–a computing approach to match similarity between higher education programs and job market demands based on maximum number of common words. International Journal of Emerging Technologies in Learning (iJET) 2022; 17(12): 258-268.

Kadhim EA., Feizi-Derakhshi MR., Aghdasi HS. Advanced text summarization model incorporating nlp techniques and feature-based scoring. IEEE Access 2025.

Lin CY. Rouge: a package for automatic evaluation of summaries. In: Text Summarization Branches Out 2004; 74-81.

Liu Y., Lapata M. Text summarization with pretrained encoders. arXiv preprint arXiv:1908.08345 2019.

Loper E., Bird S. Nltk: the natural language toolkit. arXiv preprint cs/0205028 2002.

Mihalcea R., Tarau P. Textrank: bringing order into text. In: Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing 2004; 404-411.

Mihalcea R. Graph-based natural language processing and information retrieval. Cambridge University Press 2011.

Naing I., Funabiki N., Wai KH., Aung ST. A design of automatic reference paper collection system using Selenium and Bert model. In: 2023 IEEE 12th Global Conference on Consumer Electronics (GCCE) 2023; 267-268.

Nenkova A., McKeown K. Automatic summarization. Foundations and Trends® in Information Retrieval 2011; 5(2–3): 103-233.

Orel E., Ciglenecki I., Thiabaud A., Temerev A., Calmy A., Keiser O., Merzouki A. An automated literature review tool (LiteRev) for streamlining and accelerating research using natural language processing and machine learning: descriptive performance evaluation study. Journal of Medical Internet Research 2023; 25: e39736.

Papineni K., Roukos S., Ward T., Zhu WJ. Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics 2002; 311-318.

Ramos J. Using tf-idf to determine word relevance in document queries. In: Proceedings of the First Instructional Conference on Machine Learning 2003; 242(1): 29-48.

See A., Liu PJ., Manning CD. Get to the point: summarization with pointer-generator networks. arXiv preprint arXiv:1704.04368 2017.

Setiawan GH., Adnyana IM. Improving helpdesk chatbot performance with term frequency-inverse document frequency (TF-IDF) and cosine similarity models. Journal of Applied Informatics and Computing 2023; 7(2): 252-257.

Sofi SM., Selamat A. Aspect based sentiment analysis: feature extraction using latent dirichlet allocation (LDA) and term frequency-inverse document frequency (TF-IDF) in machine learning (ML). Malaysian Journal of Information and Communication Technology (MyJICT) 2023; 169-179.

Thirumahal R. TF-IDF vectorization and clustering for extractive text summarization. Journal of Information Technology and Digital World 2024; 6(1): 96-111.

Vaswani A. Attention is all you need. In: Advances in Neural Information Processing Systems 2017.