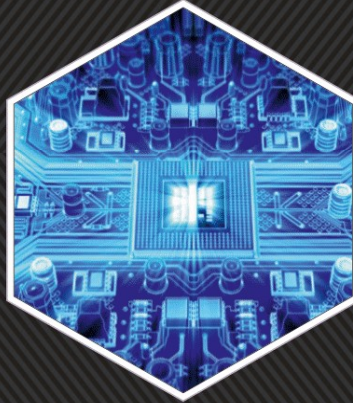




BİLGİSAYAR BİLİMLERİ VE TEKNOLOJİLERİ DERGİSİ

JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGIES



EDİTÖR: DOÇ. DR. Erdiñç AVAROĞLU
ISSN 2717 - 8579



Bilgisayar Bilimleri ve Teknolojileri Dergisi

BİLGİSAYAR BİLİMLERİ VE TEKNOLOJİLERİ DERGİSİ

CİLT 4, SAYI 1

ISSN: 2717-8579

HAZİRAN 2023



Bilgisayar Bilimleri ve Teknolojileri Dergisi

Dergi Hakkında

Bilgisayar Bilimleri ve Teknolojileri Dergisi bilim ve teknolojideki gelişmelere paralel olarak bilgisayar bilimleri ve teknolojileri alanında yeni gelişmelerle ilgili yapılan çalışmaları yayınlayan bir dergidir.

Amaç & Kapsam

BIBTED Dergisi,

✚ Bilgisayar Bilimleri ve Teknolojileri Dergisinin amacı bilgisayar alanında yapılan özgün çalışmaları yayınlamaktır. Yazım kurallarına uygun olarak hazırlanan eser, dergi editörlüğünce değerlendirme için hakemlere gönderilir. Bilgisayar Bilimleri ve Teknolojileri Dergisinde **KÖR HAKEMLİK** uygulaması mevcuttur. Yayımlanmasına, hakemlerin görüşü doğrultusunda Dergi Editör ve Yayın Kurulu karar verir. Gönderilen makaleler yayınlansın veya yayınlansın iade edilmez. Dergimizde yayımlanan yazıların her türlü sorumluluğu (bilimsel, mesleki, hukuki, etik vb.) yazarlara aittir. Yayımlanan yazıların telif hakkı dergiye aittir ve referans gösterilmeden aktarılamaz. Araştırmacılar arasındaki bilimsel iletişimi oluşturmak amacıyla aşağıda nitelikleri açıklanan, başka bir yerde yayımlanmamış makaleler Türkçe ve İngilizce olarak kabul edilmekte ancak Türkçe Kabul edilen makalenin özetinin İngilizce de basılması zorunluluğu vardır.

Aşağıdaki türlerdeki makaleler dergide yayına kabul edilmektedir:

- ✚ **Araştırma makalesi:** Özgün bir araştırmayı sonuçlarıyla birlikte sunan makale,
- ✚ **Derleme makale:** Bilgisayar Mühendisliği alanında belli bir konuda yeterli sayıda bilimsel makaleyi tarayıp, özetleyen, değerlendirme yapan ve bulguları yorumlayan makale,
- ✚ **Endüstriyel makale:** Bu alanda endüstride yapılan araştırma ve geliştirilen yeni ürün veya teknolojilerin açıklandığı makale,
- ✚ **Tez çalışması:** Lisansüstü düzeyde yapılan özgün bir tez çalışmasının genişletilmiş özetini içeren yazı,
- ✚ **Kitap yorumu:** Bilgisayar mühendisliği alanında yayımlanmış yeni bir kitabın tanıtılması ve değerlendirilmesi.
- ✚ **Kısa Bildiri:** Yapılan bir araştırmanın önemli bulgularını açıklayan yeni bir yöntem veya teknik tanımlayan yazılar.

Bütün yazıların Telif Hakkı Devri, yazarlarına bir form gönderilmek suretiyle alınır. Telif Hakkı Devir Formu göndermeyen yazarların yayımları işleme konmaz. Yayımlanmasına karar verilen yazılar üzerine yazarlarınca hiçbir eklenti yapılamaz.

Her yazı konusu ile ilgili en az iki hakeme gönderilerek şekil ve içerik bakımından incelettilir. Dergide yayımlanabilecek nitelikteki yazılar dizgisi yapıldıktan sonra, yazarlarına gönderilerek baskı öncesi gözden istenir. Makale içinde, dergide basıldığı haliyle gözükken hataların sorumluluğu yazarlarına aittir. Hata, editörlük ofisinden kaynaklandığı takdirde düzeltme yayımlanabilir.

Derginin Kapsamı;

Bilgisayar Bilimleri ve Teknolojileri Dergisinin kapsamı, akıllı sistemler, algoritmalar, benzetim, bilgisayar ağları, bilgisayar grafiği, bilgisayarla görme, bilgisayar mimarisi, bilgiye erişim, bilimsel hesaplama, bilişim güvenliği, biyoenformatik, kriptografi, paralel işleme, doğal dil işleme donanım, görüntü işleme, hesaplama kuramı, işaret işleme, işletim sistemleri, makine öğrenmesi, mobil sistemler, modelleme, tıbbi bilişim, veri madenciliği, veri tabanı sistemleri, yazılım mühendisliği, siber güvenlik, yapay zeka dahil olmak üzere bilgisayar bilimleri ve teknolojilerin tüm alanları içerir.

Yayımlanma Sıklığı

Yılda 2 sayı

ISSN

2717-8579

WEB

<https://dergipark.org.tr/tr/pub/bibted>

İletişim

eavaroglu@mersin.edu.tr / ttuncer@firat.edu.tr / kemaladem@gmail.com



Bilgisayar Bilimleri ve Teknolojileri Dergisi

EDİTÖR

Doç. Dr. Erdinç AVAROĞLU

Mersin Üniversitesi, Mühendislik Fakültesi / Bilgisayar Mühendisliği, Mersin

EDİTÖR YARDIMCILARI

Doç. Dr. Taner TUNCER

Fırat Üniversitesi, Mühendislik Fakültesi / Bilgisayar Mühendisliği, Elâzığ

Dr. Öğr. Üyesi. Kemal ADEM

Aksaray Üniversitesi, İktisadi ve İdari Bilimler Fakültesi / Yönetim Bilişim Sistemleri, Aksaray

EDİTÖR KURULU

- **Prof. Dr. Zeki YETKİN, MERSİN ÜNİVERSİTESİ**
- **Doç. Dr. İsmail KOYUNCU, AYFON KOCATEPE ÜNİVERSİTESİ**
- **Dr. Öğr. Üyesi Murat TUNA, KIRKLARELİ ÜNİVERSİTESİ**
- **Dr. Öğr. Üyesi Abdullah ELEWİ, MERSİN ÜNİVERSİTESİ**
- **Dr. Öğr. Üyesi Abdullah Erhan AKKAYA, İNÖNÜ ÜNİVERSİTESİ**
- **Dr. Öğr. Üyesi Lütfiye KUŞAK, MERSİN ÜNİVERSİTESİ**
- **Dr. Öğr. Üyesi Fatma Bünyal ÜNEL, MERSİN ÜNİVERSİTESİ**
- **Dr. Öğr. Üyesi Çiğdem ACI, MERSİN ÜNİVERSİTESİ**
- **Dr. Öğr. Üyesi Soner KIZILOLUK, TURGUT ÖZAL ÜNİVERSİTESİ**
- **Dr. Öğr. Üyesi Selman YAKUT, TURGUT ÖZAL ÜNİVERSİTESİ**

DANIŞMA KURULU

- **Prof. Dr. Ahmet Bedri ÖZER, FIRAT ÜNİVERSİTESİ**
- **Prof. Dr. Murat YAKAR, MERSİN ÜNİVERSİTESİ**
- **Doç. Dr. Fatih ÖZKAYNAK, FIRAT ÜNİVERSİTESİ**
- **Dr. Öğr. Üyesi Mehmet ACI, MERSİN ÜNİVERSİTESİ**
- **Dr. Öğr. Üyesi Murat TUNA, KIRKLARELİ ÜNİVERSİTESİ**
- **Doç. Dr. İsmail KOYUNCU, AFYON KOCATEPE ÜNİVERSİTESİ**

DİL EDİTÖRLERİ

- **Dr. Öğr. Üyesi Abdullah ELEWİ, MERSİN ÜNİVERSİTESİ**
- **Dr. Öğr. Üyesi Abdullah Erhan AKKAYA, İNÖNÜ ÜNİVERSİTESİ**
- **Arş. Gör. Dr. Dilek SABANCI, GAZİOSMANPAŞA ÜNİVERSİTESİ**

MİZANPAJ

- **Arş. Gör. Semih KAHVECİ, MERSİN ÜNİVERSİTESİ**
- **Arş. Gör. Ramazan AKKURT, MERSİN ÜNİVERSİTESİ**



Bilgisayar Bilimleri ve Teknolojileri Dergisi

İçindekiler

Contents

ARAŞTIRMA MAKALELERİ; **RESEARCH ARTICLES;**

S.No

-
- | | |
|-------|---|
| 01-07 | <i>Yüz İfadelerini Sınıflandırmada CNN Modellerinde Kullanılan Optimizasyon Yöntemlerinin Karşılaştırılması</i>
<i>Comparison of Optimization Methods Used in CNN Models for Classification of Facial Expressions</i>
Berrin İŞLEK, Hamza EROL |
| 08-18 | <i>A Natural Language Processing-Based Turkish Diagnosis Recommendation System</i>
<i>Doğal Dil İşleme Tabanlı Türkçe Tanı Öneri Sistemi</i>
Özlem Özcan KILIÇSAYMAZ, Servet BADEM |
| 19-26 | <i>An Extractive Text Summarization Model for Generating Extended Abstracts of Medical Papers in Turkish</i>
<i>Tıp Makalelerinin Genişletilmiş Özetlerini Oluşturmak İçin Çıkarımsal Bir Türkçe Metin Özetleme Modeli</i>
Anıl KUŞ, Çiğdem İnan ACI |
-



Araştırma Makalesi

Yüz İfadelerini Sınıflandırmada CNN Modellerinde Kullanılan Optimizasyon Yöntemlerinin Karşılaştırılması

Berrin İŞLEK*¹, Hamza EROL²

¹Sivas Bilim ve Teknoloji Üniversitesi, Mühendislik ve Doğa Bilimleri Fakültesi, Bilgisayar Mühendisliği Bölümü, Sivas, Türkiye

²Mersin Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, Mersin, Türkiye

ÖZ

Anahtar Kelimeler:

Evrişimli Sinir Ağı
Optimizasyon Yöntemleri
Yüz İfadeleri
Sınıflandırma
Veri Arttırma

İnsan yüz ifadeleri, kişiler için iletişimde ana bilgi kanallarından biridir. İnsanlar günlük hayatlarında psikolojik durumları ifade etmek için çok fazla yüz ifadesi oluşturmaktadır. Bu yüz ifadeleri temel ve karmaşık duygular olarak ayrılır. İnsanlar bu duygu ifadelerini tanımlamada hala zorlanırken makineler için de geliştirmekte olan bir konudur. Bu sebeple son zamanlarda çok fazla ilgi görmektedir. Bu çalışmada Ohio Eyalet Üniversitesine ait Compound Emotion (CE) veri setindeki temel 7 duygu olan doğal, mutlu, üzgün, öfkeli, şaşırılmış, korkulu ve iğrenme duyguları üzerinde durulmaktadır. Veri seti 1610 görüntüden oluşmaktadır. Başarımı arttırmak için veri çoğaltma işlemi uygulanarak 5478 görüntü elde edilmektedir. Eğitilmiş Evrişimli sinir ağı (CNN) modelleri ile sınıflandırma işleminde optimizasyon yöntemlerinin etkileri gösterilmektedir. VGG19 ve MobileNet modelleri ile Adadelta, Adagrad ve Stokastik gradyan inişi (SGD) optimizasyon yöntemlerinin duygu sınıfları üzerindeki sonuçları grafikler ve değerlendirme tabloları ile birlikte detaylı incelenmektedir.

Comparison of Optimization Methods Used in CNN Models for Classification of Facial Expressions

Keywords:

Convolutional Neural Network
Optimization Methods
Face Expressions
Classification
Data Augmentation

ABSTRACT

Human facial expressions are one of the main channels of communication for people. People make too many facial expressions to express psychological states in their daily lives. These facial expressions are divided into basic and complex emotions. While humans still struggle to identify these expressions of emotion, it is an emerging topic for machines as well. For this reason, it has attracted a lot of attention lately. In this study, the 7 basic emotions in the Compound Emotion (CE) dataset of Ohio State University, which are natural, happy, sad, angry, surprised, fearful and disgusted, are emphasized. The dataset consists of 1610 images. To increase the performance, 5478 images are obtained by applying the data duplication process. The effects of optimization methods in the classification process are shown with trained convolutional neural network (CNN) models. The results of VGG19 and MobileNet models and Adadelta, Adagrad and Stochastic gradient descent (SGD) optimization methods on emotion classes are examined in detail with graphics and evaluation tables.

*Sorumlu Yazar

*(berrinislek@sivas.edu.tr) ORCID ID 0000 - 0003 - 1984 - 357X
(herol@mersin.edu.tr) ORCID ID 0000 - 0001 - 8983 - 4797

e-ISSN: 2717-8579

Geliş Tarihi: 18/11/2022; Kabul Tarihi: 24/02/2023

Bilgisayar Bilimleri ve Teknolojileri Dergisi

1. GİRİŞ

Yüz ifadelerinin oluşturduğu duygular, yüzyıllardır psikoloji ve bilişim alanında önemli araştırma konusudur. Bir kişinin yüz ifadelerinin analizi ile fiziksel ve duygusal durumu ile ilgili birçok çıkarım yapmak mümkündür. Günümüz teknolojik gelişmeleriyle beraber bireylerin yüz ifadelerinden duygu tespitinin yapılması medikal (Yolcu vd., 2017), robotik (Littlewort vd., 2003), trafik (Zhang ve Hua, 2015), pazarlama gibi farklı alanlarda kullanımı yaygınlaşmıştır.

Araştırmalarda duygular, temel ve karmaşık duygular olarak iki sınıfta incelenmektedir. Araştırmalar iki duygu sınıfı olduğunu gösterse de çalışmalar genellikle temel duyguların tespiti üzerine yoğunlaştığı görülmektedir. Temel duygular mutlu (happy), üzgün (sad), öfkeli (angry), şaşırılmış (surprised), iğrenmiş (disgusted), doğal (neutral) ve korkulu (fearful) yüz ifadeleridir. Yüz ifadelerinin belirlenmesinin temel hedef, belirli yüz görünümüne karşılık gelen insanların duygu durumunu tanımlamaktır. Bu nedenle insan bilgisayar etkileşimi olan birçok sosyal uygulamalarda sıkça kullanılmaktadır (Ko, 2018).

Önceki yıllarda, birçok geleneksel yöntemlerle ön işlem ve özellik çıkarım ile kullanılan makine öğrenmesi yöntemi önerilmiştir. Ancak birçok sebeple istenilen başarımlar elde edilmemektedir (Li vd., 2020). Çoklu hesaplama modellerine izin veren yapısı, büyük ve karmaşık verileri öğrenmesinde verimli sonuçlar vermesinden dolayı yüz ifadelerinden duygu analizi için derin öğrenme yöntemlerinin kullanımı artmaktadır ve yüksek başarımlar elde edilen birçok çalışma yapılmıştır (Voulodimos ve Doulamis, 2018).

2015 yılında Chen ve arkadaşları, görüntü verisinden duygu analizi için oluşturdukları Evrişimli Sinir Ağı (CNN) modeli geleneksel yöntemlerden daha iyi sonuçlar elde etmiştir (Chen vd., 2015). Karaman ve Özdemir, bir CNN modeli olan Alexnet ile video karelerinden yüz ifadeleri tanıma sistemi oluşturmuştur (Özdemir ve Karaman, 2017). Jung ve arkadaşları, görüntü dizilerinden zamansal görünüm özellikleri ve zamansal geometri özellikleri çıkartan iki farklı CNN modelini birleştirerek daha yüksek başarımlar elde etmeye çalışmışlardır (Jung vd., 2015).

Videla ve Kumar arkadaşları, 10 katmanlı CNN modeli ve Adam optimizasyon yöntemi tercih edilerek Cohn-Kanade (CK+) ve Japon Kadın Yüz İfadeleri (JAFFE) veri setleri üzerinde yüz ifadeleri tespiti üzerinde durulmuştur (Videla ve Kumar, 2020).

2022 yılında Kandhro ve arkadaşları, CK+ ve JAFFE veri setlerini kullanarak CNN modeli üzerinde hiperparametrelerin etkileri incelenmiştir. Adamax, Nadam ve Adam gibi optimizasyon yöntemleri test edilmiştir (Kandhro vd., 2022).

En önemli derin öğrenme algoritmalarından biri evrişimli sinir ağı (CNN) modelidir. Derin öğrenme modellerinin performans ve başarımları için

optimizasyon algoritması seçimi önemli bir parametredir. Yapılan çalışma kapsamında VGG19 ve MobileNet modelleri SGD optimizasyon algoritması, Adadelta optimizasyon algoritması ve Adagrad optimizasyon algoritmaları kullanılarak birbirleriyle detaylı olarak karşılaştırılmış ve başarımları incelenmiştir.

Makalenin bir sonraki kısmı olan ikinci kısımda çalışmada kullanılan veri seti, modeller ve optimizasyon yöntemlerinden bahsedilmektedir. Daha sonra üçüncü kısım ise kullanılan parametreler, elde edilen grafikler ve değerlendirme tabloları yer almaktadır. Son olarak dördüncü kısımda sonuçların değerlendirilmesi verilmektedir.

2. YÖNTEM

Bu çalışmada, Ohio Eyalet Üniversitesine ait Compound Emotion (CE) veri seti kullanılmaktadır. Yaşları 23 olan 130'u kadın 100'ü erkek 230 denek kişiden oluşmaktadır. Kişiler beş farklı kökenden oluşmaktadır. Yüz hatlarının belirginliği için sakal ile gözlük bulunmamaktadır. Ayrıca kaşlarının belirginliği belli olması için alınlarını açmaları istenmiştir. Bu veri seti temel ve karmaşık duyguları kapsamaktadır. İnsanlar günlük hayatlarında birçok duyguyu tanısalar da çalışmalar genellikle 7 temel duyguyu esas almıştır. Bu çalışmamızda veri setinde bulunan temel duygular olan mutlu, üzgün, öfkeli, şaşırılmış, doğal, iğrenmiş ve korkulu yüz ifadeleri kullanılmıştır. Toplam 1610 görüntü bulunmaktadır (Du vd., 2014). Veri seti %70 eğitim ve %30 test olarak ayrılmıştır. Veri setinden örnek görüntüler Şekil 1'de gösterilmektedir (Du vd., 2014).



Şekil 1. Yüz ifadelerinden (a) doğal, (b) mutlu, (c) üzgün, (d) öfkeli, (e) şaşırılmış, (f) iğrenmiş, (g) korkulu için veri setinden örnek görüntüler

Veri seti boyutu derin öğrenme modelleri için önemli bir parametredir. Probleme uygun veri toplamak zor bir işlemdir. Veri seti birleştirme ve veri çoğaltma işlemleri bu sorun için çözüm olabilmektedir. Veri seti birleştirme işleminde görüntülerde uyumsuzluk göstermesi sorun olabilmektedir. Bu sebeple modellerin başarımlarını arttırmak ve aşırı öğrenmenin önüne geçmek için eğitim veri setine veri arttırma işlemi uygulanmıştır (Alimovski ve Erdemir, 2021). Bu işlem sadece veri setinde eğitim için ayrılan kısma uygulanmaktadır. Mevcut verilerden yeni görüntüler üretilmektedir. Uygulanan veri arttırma yöntemi, Bir derin öğrenme

kütüphanesi olan Tensorflow'un açık kaynak olarak bulunan veri artırma kodundan geliştirildi (Tensorflow Core, 2020). Görüntü üzerinde sağa ve sola belli açılarda döndürme, yakınlaştırma ve ölçekleme işlemleri yapıldı. Bu işlemler sonucunda toplam eğitim setindeki görüntü sayısı 5478 olmuştur. Test veri setinde bir değişiklik yapılmamıştır. Test veri setinde altı duygu için 69 ve üzgün sınıfında 70 veri ile 483 görüntü bulunmaktadır.

Bu çalışmada, önceden eğitilmiş Evrişimli Sinir Ağları (CNN) modelleri olan VGG19 ve MobileNet modelleri tercih edilmiştir. VGG19 modeli, çok katmanlı derin bir sinir ağıdır. VGG19 modelinde maksimum havuzlama katmanları bulunmaktadır. 4096 nörondan oluşan iki tam bağlantı katmanı içermektedir. 19 katman derinliğine sahiptir (Zheng vd., 2018). MobileNet modeli, mobil uygulamalarda kullanılmak için TensorFlow'un ilk mobil derin öğrenme modelidir. MobileNet katmanları derinlemesine ayrılabilir evrişimlerden oluşur. Parametre sayısını önemli derece azaltmaktadır (Pujara, 2020). Optimizasyon yöntemleri, makine öğrenmesi ve derin öğrenme yöntemlerinde hata oranını en aza indirmek için önemlidir. Bu çalışmada literatürden farklı olarak üç optimizasyon yöntemi seçilmiştir. Bunlar Stokastik gradyan inişi (SGD), Adadelta ve Adagrad yöntemleridir (Seyyarer vd., 2020; Defazio, 2020).

3. BULGULAR

Çalışma kapsamında temel 7 duygu olan mutlu (happy), üzgün (sad), öfkeli (angry), şaşırılmış (surprised), iğrenmiş (disgusted), doğal (neutral) ve korkulu (fearful) yüz ifadelerinin belirlenmesi için VGG19 ve MobileNet modelleri ile birlikte SGD, Adadelta ve Adagrad optimizasyon yöntemleri kullanılmaktadır. Modellere etki eden hiperparametreler test edilerek seçilmiştir. Bunlardan dönem sayısı(epoch) 10, parti boyutu (batch size) 32, öğrenme oranı (learning rate) 0.001 ve girdi boyutu (224,224,3) belirlenip her model için sabit değerlerdir. Optimizasyon algoritmalarının ve modellerin başarımlarının doğru kıyaslanması için başarımlar (accuracy) grafiği, hata (loss) grafiği, karmaşıklık (confusion) matrisi ve sınıfların performans metrikleri sunulmaktadır. Test verisi 484 görüntüden oluşmaktadır. Sonuçlar bu görüntüler üzerinden elde edilmektedir.

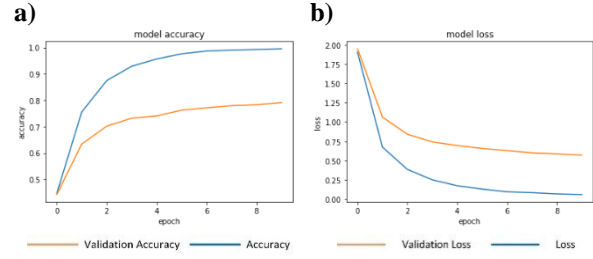
3.1. VGG19 Modeli ile Elde Edilen Bulgular

VGG19 CNN modeli için üç farklı optimizasyon yöntemi ayrı ayrı incelenmekte ve birbirleri arasında kıyaslanmaktadır.

3.1.1. SGD Optimizasyon Yöntemi

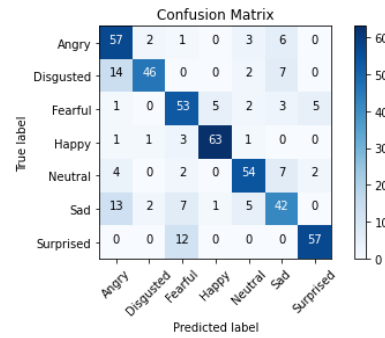
Stokastik Gradyan İnişi (SGD) optimizasyon yöntemi ile doğrulama ve eğitim verileri için başarımlar

(accuracy) ve hata (loss) grafikleri şekil 2'de sunulmaktadır.



Şekil 2. Vgg19 modeli ile kullanılan SGD optimizasyon algoritmasının a) başarımlar grafiği b) hata grafiği

Şekil 2'de grafikler paralel ilerlemiş olsa da değerler arasında farklar bulunmaktadır.



Şekil 3. Vgg19 modeli ile kullanılan SGD optimizasyon algoritmasının karmaşıklık matrisi

Şekil 3 ile test verileri için 7 duygu sınıfının doğru tahmin ettiği değerler görülmektedir. En çok doğru tahmini mutlu duygu sınıfında yapmıştır. En düşük tahmin ise üzgün duygu sınıfında olmuştur.

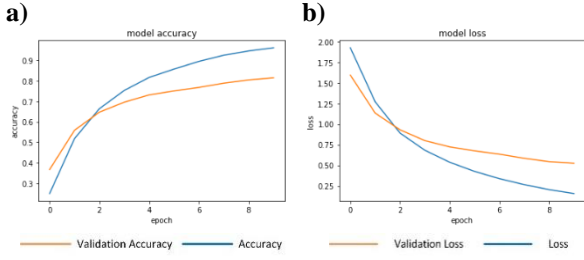
Tablo 1. Vgg19 model ile SGD optimizasyon algoritması için performans sonuçları

	PRECISION	RECALL	F1-SCORE	SUPPORT
ANGRY	0.63	0.83	0.72	69
DISGUSTED	0.90	0.67	0.77	69
FEARFUL	0.68	0.77	0.72	69
HAPPY	0.91	0.91	0.91	69
NEUTRAL	0.81	0.78	0.79	69
SAD	0.65	0.60	0.62	70
SURPRISED	0.89	0.83	0.86	69
ACCURACY			0.77	484
MACRO AVG	0.78	0.77	0.77	484
WEIGHTED	0.78	0.77	0.77	484
AVG				

Tablo 1'de test başarımlar (accuracy) değerinin %77 olduğu görülmektedir.

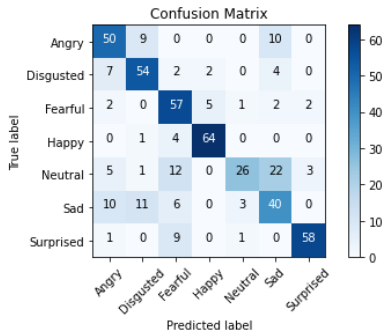
3.1.2. Adadelta Optimizasyon Yöntemi

Adadelta optimizasyon algoritması için doğrulama ve eğitim verileri için başarımlar ve hata grafikleri şekil 4 ile sunulmaktadır.



Şekil 4. Vgg19 modeli ile kullanılan Adadelta optimizasyon algoritmasının a) başarım grafiği b) hata grafiği

Şekil 4'te başarım ve doğrulama değerleri benzer seyretmektedir.



Şekil 5. Vgg19 modeli ile kullanılan Adadelta optimizasyon algoritmasının karmaşıklık matrisi

Şekil 5'te test verileri için 7 duygu sınıfının doğru tahmin ettiği değerler görülmektedir. En çok doğru tahmini mutlu duygu sınıfında yapmıştır. En düşük tahmin ise doğal duygu sınıfında olmuştur.

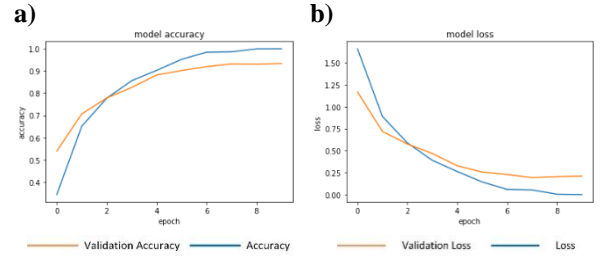
Tablo 2. Vgg19 model ile Adadelta optimizasyon algoritması için performans sonuçları

	PRECISION	RECALL	F1-SCORE	SUPPORT
ANGRY	0.67	0.72	0.69	69
DISGUSTED	0.71	0.78	0.74	69
FEARFUL	0.63	0.83	0.72	69
HAPPY	0.90	0.93	0.91	69
NEUTRAL	0.84	0.38	0.52	69
SAD	0.51	0.57	0.54	70
SURPRISED	0.92	0.84	0.88	69
ACCURACY			0.72	484
MACRO AVG	0.74	0.72	0.72	484
WEIGHTED	0.74	0.72	0.72	484
AVG				

Tablo 2'de test başarım (accuracy) değerinin %72 olduğu görülmektedir.

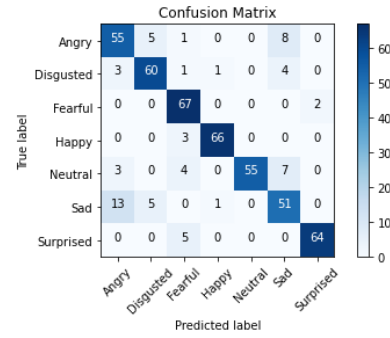
3.1.3. Adagrad Optimizasyon Yöntemi

Adagrad optimizasyon algoritması için doğrulama ve eğitim verileri için başarım ve hata grafikleri şekil 6 ile sunulmaktadır.



Şekil 6. Vgg19 modeli ile kullanılan Adagrad optimizasyon algoritmasının a) başarım grafiği b) hata grafiği

Şekil 6'da başarım ve doğrulama verileri uyumlu ve paralel seyretmektedir.



Şekil 7. Vgg19 modeli ile kullanılan Adagrad optimizasyon algoritmasının karmaşıklık matrisi

Şekil 7 ile test verileri için 7 duygu sınıfının doğru tahmin ettiği değerler görülmektedir. Genel olarak üzgün duygu sınıfı dışında iyi sonuçlar vermiştir.

Tablo 3. Vgg19 model ile Adagrad optimizasyon algoritması için performans sonuçları

	PRECISION	RECALL	F1-SCORE	SUPPORT
ANGRY	0.74	0.80	0.77	69
DISGUSTED	0.86	0.87	0.86	69
FEARFUL	0.83	0.97	0.89	69
HAPPY	0.97	0.96	0.96	69
NEUTRAL	1.00	0.80	0.89	69
SAD	0.73	0.73	0.73	70
SURPRISED	0.97	0.93	0.95	69
ACCURACY			0.86	484
MACRO AVG	0.87	0.86	0.86	484
WEIGHTED	0.87	0.86	0.86	484
AVG				

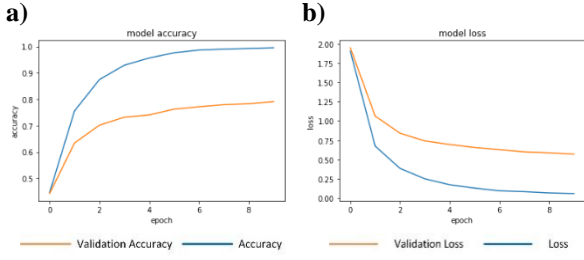
Tablo 3'te test başarım (accuracy) değerinin %86 olduğu görülmektedir.

3.2. MobileNet Modeli ile Elde Edilen Bulgular

MobileNet modeli için üç farklı optimizasyon yöntemi ayrı ayrı incelenmekte ve birbiri arasında kıyaslanmaktadır.

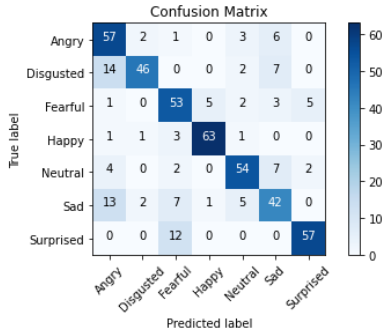
3.2.1. SGD Optimizasyon Yöntemi

SGD optimizasyon algoritması için doğrulama ve eğitim verileri için başarım ve hata grafikleri şekil 8 ile sunulmaktadır.



Şekil 8. MobileNet modeli ile kullanılan SGD optimizasyon algoritmasının a) başarımlar grafiği b) hata grafiği

Şekil 8'te eğitim ve doğrulama verileri arasında başarımlar farkları görülmektedir.



Şekil 9. MobileNet modeli ile kullanılan SGD optimizasyon algoritmasının karmaşıklık matrisi

Şekil 9'da test verileri için 7 duygu sınıfının doğru tahmin ettiği değerler görülmektedir. 69 test verisinden 63'ünü doğru tahmin ettiği en iyi sonuç mutlu yüz ifadesidir.

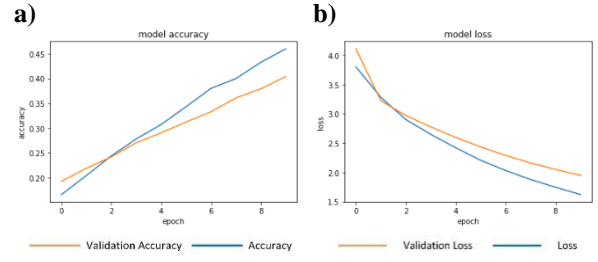
Tablo 4. MobileNet model ile SGD optimizasyon algoritması için performans sonuçları

	PRECISION	RECALL	F1-SCORE	SUPPORT
ANGRY	0.63	0.83	0.72	69
DISGUSTED	0.90	0.67	0.77	69
FEARFUL	0.68	0.77	0.72	69
HAPPY	0.91	0.91	0.91	69
NEUTRAL	0.81	0.78	0.79	69
SAD	0.65	0.60	0.62	70
SURPRISED	0.89	0.83	0.86	69
ACCURACY			0.77	484
MACRO AVG	0.78	0.77	0.77	484
WEIGHTED AVG	0.78	0.77	0.77	484

Tablo 4 ile test başarımlar (accuracy) değerinin %77 olduğu görülmektedir.

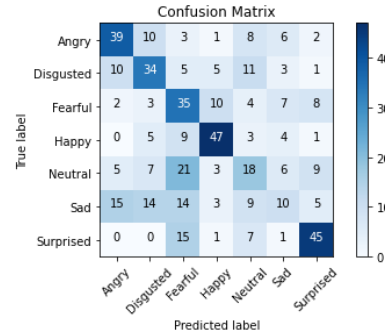
3.2.2. Adadelta Optimizasyon Yöntemi

Adadelta optimizasyon algoritması için doğrulama ve eğitim verileri için başarımlar ve hata grafiğleri Şekil 10 ile sunulmaktadır.



Şekil 10. MobileNet modeli ile kullanılan Adadelta optimizasyon algoritmasının a) başarımlar grafiği b) hata grafiği

Şekil 10'da başarımlar ve doğrulama verileri paralel ilerlemiştir. Ancak başarımlar düşük bulunmaktadır. Hata değeri ise yüksek bulunmaktadır.



Şekil 11. MobileNet modeli ile kullanılan Adadelta optimizasyon algoritmasının karmaşıklık matrisi

Şekil 11 ile gösterilen matriste test verileri için 7 duygu sınıfının doğru tahmin ettiği değerler görülmektedir. 69 test verisinden 47'sini doğru tahmin ettiği en iyi sonuç mutlu yüz ifadesidir. 45 doğru tahmin ile şaşırmiş duygu sınıfı gelmektedir.

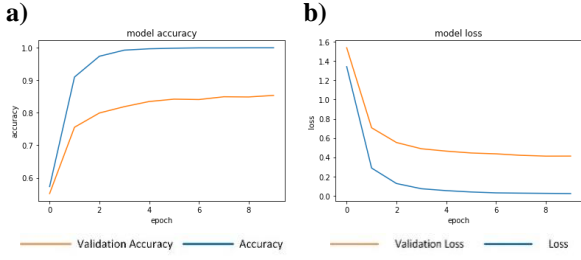
Tablo 5. MobileNet model ile Adadelta optimizasyon algoritması için performans sonuçları

	PRECISION	RECALL	F1-SCORE	SUPPORT
ANGRY	0.55	0.57	0.56	69
DISGUSTED	0.47	0.49	0.48	69
FEARFUL	0.34	0.51	0.41	69
HAPPY	0.67	0.68	0.68	69
NEUTRAL	0.30	0.26	0.28	69
SAD	0.27	0.14	0.19	70
SURPRISED	0.63	0.65	0.64	69
ACCURACY			0.47	484
MACRO AVG	0.46	0.47	0.46	484
WEIGHTED AVG	0.46	0.47	0.46	484

Tablo 5 ile test başarımlar (accuracy) değerinin %47 olduğu görülmektedir.

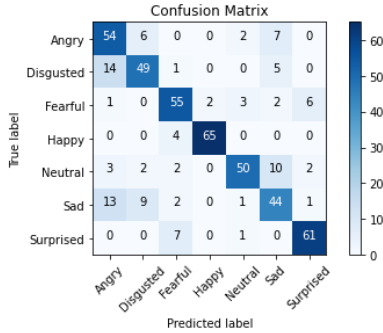
3.2.3. Adagrad Optimizasyon Yöntemi

Adagrad optimizasyon algoritması için doğrulama ve eğitim verileri için başarımlar ve hata grafiğleri Şekil 12 ile sunulmaktadır.



Şekil 12. MobileNet modeli ile kullanılan Adagrad optimizasyon algoritmasının a) başarımlar grafiği b) hata grafiği

Şekil 12’de eğitim ve doğrulama verilerinde başarımlar değerleri arasında fark görülmektedir.



Şekil 13. MobileNet modeli ile kullanılan Adagrad optimizasyon algoritmasının karmaşıklık matrisi

Şekil 13 ile gösterilen matriste test verileri için 7 duygu sınıfının doğru tahmin ettiği değerler görülmektedir. 69 test verisinden 65’ini doğru tahmin ettiği en iyi sonucu mutlu yüz ifadesidir.

Tablo 6. MobileNet model ile Adagrad optimizasyon algoritması için performans sonuçları

	PRECISION	RECALL	F1-SCORE	SUPPORT
ANGRY	0.64	0.78	0.70	69
DISGUSTED	0.74	0.71	0.73	69
FEARFUL	0.77	0.80	0.79	69
HAPPY	0.97	0.94	0.96	69
NEUTRAL	0.88	0.72	0.79	69
SAD	0.65	0.63	0.64	70
SURPRISED	0.87	0.88	0.88	69
ACCURACY			0.78	484
MACRO AVG	0.79	0.78	0.78	484
WEIGHTED AVG	0.79	0.78	0.78	484

Tablo 6 ile test başarımlar (accuracy) değerinin %78 olduğu görülmektedir.

4. SONUÇLAR

Yapılan çalışmada, yüz ifadelerinden oluşan görüntü veri seti kullanarak VGG19 ve MobileNet modellerine uygulanan SGD, Adadelta ve Adagrad optimizasyon yöntemlerinin kıyaslanması yapılmıştır. Bulgular sonucunda VGG19 modelinin en iyi sonucu %86 oranla Adagrad optimizasyon algoritması ile elde edilmiştir. En düşük sonuçlar ise %72 oranla Adadelta optimizasyon algoritması tarafından elde edilmiştir. Tüm optimizasyon

algoritmaları için en iyi sonucu mutlu duygu sınıfında elde etmiştir. MobileNet modeli en iyi sonuçları %78 ile Adagrad ve %77 ile SGD optimizasyon algoritmalarında elde etmiştir. MobileNet ile kullanılan Adadelta optimizasyon yöntemi %47 oranla en düşük sonuç olmuştur.

Tüm sonuçlar değerlendirildiğinde iki model için Adagrad optimizasyon algoritması başarılı sonuç verirken en düşük değerler Adadelta optimizasyon algoritması ile elde edilmektedir. Tüm modellerde en iyi sonucu ve en çok doğru tahmini mutlu duygu sınıfı ile elde edilmektedir

KAYNAKÇA

- Alimovski, E. ve Erdemir, G. (2021). Veri artırma tekniklerinin derin öğrenmeye dayalı yüz tanıma sisteminde etkisi. *İstanbul Sabahattin Zaim Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, 3(1), 76-80.
- Chen, M., Zhang, L., & Allebach, J.P. (2015). Learning deep features for image emotion classification. *2015 IEEE International Conference on Image Processing (ICIP)*, 4491-4495.
- Defazio, A. (2020). Optimizasyon yöntemleri 1. 25 Mayıs tarihinde <https://atcold.github.io/pytorch-Deep-Learning/tr/week05/05-1/> adresinden erişildi.
- Du, S., Tao, Y. & Martinez, A. M. (2014). Compound facial expressions of emotion. *Proceedings of the national academy of sciences*, 111(15), E1454-E1462.
- Jung, H., Lee, S., Yim, J., Park, S., & Kim, J. (2015) Joint Fine-Tuning in Deep Neural Networks for Facial Expression Recognition. *2015 IEEE International Conference on Computer Vision (ICCV)*, 2983-2991. doi: 10.1109/ICCV.2015.341.
- Kandhro, I. A., Uddin, M., Hussain, S., Chaudhery, T. J., Shorfuazzaman, M., Meshref, H., ... & Khalaf, O. I. (2022). Impact of Activation, Optimization, and Regularization Methods on the Facial Expression Model Using CNN. *Computational Intelligence and Neuroscience*, 2022.
- Ko, B. (2018). A Brief Review of Facial Emotion Recognition Based on Visual Information. *Sensors*, 18(2). doi: 10.3390/s18020401.
- Li, J., Jin, K., Zhou, D., Kubota, N., & Ju, Z. (2020). Attention mechanism-based CNN for facial expression recognition. *Neurocomputing*, 411,340-350. <https://doi.org/10.1016/j.neucom.2020.06.014>.
- Littlewort, G., Bartlett, M. S., Fasel, M. S., Chenu, J., Kanda, T., Ishiguro, H. & Movellan, J. R. (2003). Towards social robots: Automatic evaluation of human-robot interaction by face detection and expression classification, 2003.

- Özdemir, D., ve Karaman, S. (2017). Investigating interactions between students with mild mental retardation and humanoid robot in terms of feedback types. *Egitim ve Bilim*, 42(191),109-138.
- Pujara, A. (2020). Image classification with mobilenet. 27 Mayıs tarihinde <https://medium.com/analytics-vidhya/image-classification-with-mobilenet-cc6fbb2cd470> adresinden erişildi.
- Seyyarer, E., Ayata, F., Uçkan, T. & Karıcı, A. (2020). Derin öğrenmede kullanılan optimizasyon algoritmalarının uygulanması ve kıyaslanması. *Computer Science*, 5(2), 90-98.
- TensorFlow Core (Kasım, 2020) https://www.tensorflow.org/api_docs/python/tf/keras/preprocessing/image/ImageDataGenerator.
- Videla, L. S., & Kumar, P. A. (2020, July). Facial expression classification using vanilla convolution neural network. In *2020 7th international conference on smart structures and systems (ICSSS)* (pp. 1-5). IEEE.
- Voulodimos, A. ve Doulamis, N. (2018). Anastasios Doulamis, Eftychios Protopapadakis. Deep Learning for Computer Vision: A Brief Review. *Computational Intelligence and Neuroscience*. <https://doi.org/10.1155/2018/7068349>
- Yolcu, G., Oztel, I., Kazan, S., Oz, C., Palaniappan, K., Lever, T. E., & Bunyak, F. (2017). Deep learning-based facial expression recognition for monitoring neurological disorders. *IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 1652-1657.
- Zhang Y. & Hua, C. (2015). Driver fatigue recognition based on facial expression analysis using local binary patterns. *Opt. - Int. J. Light Electron Opt.*, 126(23), 4501-4505.
- Zheng, Y., Yang, C., & Merkulov, A. (2018). Breast cancer screening using convolutional neural network and follow-up digital mammography. In *Computational Imaging III* (Vol. 10669, p. 1066905). SPIE.



Araştırma Makalesi

A Natural Language Processing-Based Turkish Diagnosis Recommendation System

Özlem Özcan Kılıçsıymaz*¹, Servet Badem²

Keywords:

AI-Based
recommendation system,
Turkish natural language
processing,
Long short-term
memory,
Recurrent neural network,
Healthcare
recommendation system

ABSTRACT

MD-Advisor is the abbreviation of "Medical Doctor-Advisor", a novel artificial intelligence-based (AI) recommendation system in healthcare. Moreover, the health-based recommender system is a decision-making tool that recommends appropriate healthcare information to patients and clinicians. It aims to minimize human error in the clinic and enhance patient safety by utilizing Natural Language Processing (NLP) methods and AI to diagnose certain cases that may otherwise be overlooked. In the fast-paced world of healthcare, it is essential for physicians to quickly and accurately diagnose patients to provide effective treatment. The MD-Advisor was developed to help healthcare professionals achieve this goal by speeding up the diagnostic process and presenting all possible conditions based on patient complaints. With this project, the methods of diagnosing the patient and then recommending the examination are completed quickly. Based on the data obtained from patient complaints that indicates the current health status of the patient; data preprocessing, labeling, and deep learning modeling techniques are used. The diagnostic codes used as labels for the diagnosis recommendation were obtained as output from the Recurrent Neural Networks (RNN) model. As a result of the study, the diagnosis proposal for the patient's complaints was successfully predicted with the applied RNN model approach.

Doğal Dil İşleme Tabanlı Türkçe Tanı Öneri Sistemi

ÖZ

Anahtar Kelimeler:
Yapay Zekâ Tabanlı Öneri
Sistemi, Türkçe Doğal Dil
İşleme, Uzun Kısa Süreli
Bellek, Yinelemeli sinir ağı,
Sağlık Öneri Sistemi

MD-Advisor, sağlık hizmetlerinde yapay zekâ tabanlı bir öneri sistemi olan "tıp doktoru – danışman" ifadesinin kısaltmasıdır. Ayrıca sağlık temelli öneri sistemi, hastalara ve klinisyenlere uygun sağlık hizmeti bilgileri için önerilerde bulunan bir karar alma aracıdır. MD-Advisor Projesi, doktorların hastalara teşhis koyarken izledikleri prosedürleri hızlandırmak ve olası tüm durumları kısa sürede doktora sunmak amacıyla geliştirilmiştir. Bu proje ile hastaya teşhis konulması ve sonrasında tetkik önerilmesi süreçleri çok hızlı bir şekilde tamamlanmaktadır. Böylece hasta doğrudan tedavi aşamasına geçmektedir. Hastanın mevcut sağlık durumunu gösteren hasta şikayetlerinden elde edilen verilere dayanarak; veri ön işleme, etiketleme ve derin öğrenme modelleme teknikleri kullanılmaktadır. Teşhis önerisi için etiket olarak kullanılan teşhis kodları, Tekrarlayan Sinir Ağları (TSA) modelinden çıktı olarak elde edildi. Çalışma sonucunda uygulanan TSA modeli yaklaşımı ile hastanın şikayetlerine yönelik tanı önerisi başarılı bir şekilde tahmin edilmiştir.

* Responsible writer

(ozlemozcank@gmail.com) ORCID ID 0000-0002-7282-512X
(servetbadem@gmail.com) ORCID ID 0000-0002-9883-3056

e-ISSN: 2717-8579

Arrival Date: 31/03/2022; Acceptance Date: 30/03/2023

Journal of Computer Science and Technologies

1. INTRODUCTION

When a patient applied to a physician, approximately 100 thousand diseases are likely. The physician attempts to reduce these possibilities by taking patient's history, asking questions about their complaints, examining them, and ordering tests as required. After this process, clinicians make their decisions for diagnosis. AI-based recommendation systems, such as the MD-Advisor, aims to mimic this process by processing patient's complaints and reducing the number of potential diagnoses. These systems provide the physician with one or more of the most accurate diagnoses based on the patient's symptoms. The ultimate goal is to arrive at an accurate diagnosis in a timely and efficient manner.

These predictive technologies create an online model by integrating disease-related text and ontology features into intelligent algorithms. The use of recommendation systems has several advantages, such as reducing the workload for physicians and facilitating the inference of diagnoses made by other physicians based on patient complaints. The system trains itself with the past decisions of physicians through its learning process, offering the opportunity to compare its current recommendations with the physician's diagnosis. Furthermore, the system acts as an assistant to physicians, providing them with the control to make decisions independently.

The goal of the planned system is to minimize clinical decision error rates by presenting a diagnosis plan to the physician by integrating each phase of the program into the Electronic Healthcare Records (EHR) developed within the company. A clinical decision support system will be established that operates in harmony with the system and continuously trains itself based on physician feedback. This system will assist physicians in making critical decisions.

The data for the study was obtained from the Acıbadem Healthcare Group (AHG) hospitals. AHG is a healthcare institution offering services through various hospitals and medical centers in Turkey and all around the world.

It is widely recognized that oversights in the clinical field can result in serious problems that pose a significant risk to patient safety. Malpractices within the clinical process can lead to morbidity and even mortality. The MD-Advisor program aims to minimize human error rates in the clinic and enhance patient safety by utilizing Natural Language Processing (NLP) methods and AI to diagnose certain cases that may otherwise be overlooked.

To determine the data to be used in the project, data analysis was performed to determine the suitability of the data to be taken from the Hospital Information Management System (HIMS) screens for NLP. The EHR of the internal medicine department were used due to their comprehensive nature. The existing medical ontologies were evaluated, and the relevant classes, subclasses, and relations were

identified for use in the ontology. Entities were detected with the help of Python. Many tools and libraries, such as NLTK, SpaCy, and Scikit Learn, which were created to apply NLP techniques and solve problems, were used in the text preprocessing stage.

The patient complaint data was preprocessed to make it usable for the project. During the preprocessing stage, the text was divided into tokens, and meaningless words and unnecessary characters were removed to obtain clean, understandable data. An RNN algorithm was created as a deep learning model for diagnosis prediction. As the output of this algorithm, a diagnosis recommendation is made based on the patient's situation. The study consists of three parts: introduction, methodology, and results. In the introduction, diagnostic methods and recommendation systems are described. In the methodology section, the project workflow is outlined, including data preprocessing, modeling, ontology, and Named Entity Recognition (NER) model. The results of the project were presented and analyzed in the last part, and the study is concluded.

2. BACKGROUND

Diagnosis in medicine refers to the process of determining the nature and circumstances of a medical condition through examinations and investigations. Furthermore, it refers to the decision reached from these processes as stated in the dictionary ("Definition of Diagnosis," n.d.). In this context, diagnosis can be considered as a decision-making process under uncertainty, as stated in the article "How doctors diagnose diseases and prescribe treatments: an fMRI study of diagnostic salience" (Melo et al., 2017).

In the diagnostic process, physicians start by asking questions related to the patient's symptoms and complaints such as "What brings you here?" and "What are your complaints?". Based on the patient's responses, the physician forms an initial impression of the potential disease. Subsequently, they start examination using medical tools according to the patient's complaints and the area that causes the problem. If the problem is in the heart or lungs, they use a stethoscope; if the problem is in the ears of the patient, they use an otoscope, etc. Nevertheless, all the diagnostic tools cannot be mobile so they cannot be found in the doctor's office. If the area causing the problem is not easily accessible, imaging tests such as Magnetic Resonance Imaging (MRI) or Computed Tomography (CT) scans may be ordered. Despite advancements in investigation techniques, the process of gathering information through history taking still remains an important aspect of diagnosis. Nowadays, investigation techniques are improved, and doctors can detect diseases easier than ever. However, the part where doctors ask questions to their patients about the disease remains the same. Therefore, in this project, the focus is on utilizing

EHR to suggest possible diagnoses to physicians. EHRs capture the information recorded by physicians during patient interviews and are a valuable resource for diagnosis decision-making.

Recommender systems are designed to support medical care providers, particularly physicians, in the selection of appropriate diagnoses, treatments, medications, and other recommendations they may require. (Stark, Knahl, Aydin, & Elish, 2019) In recent decades, substantial amounts of data have been accumulated in clinical databases, including patients' health information, laboratory results, medical reports, treatment plans, and physicians' notes. As a result, the availability of digital information for patient-oriented decision-making applications has significantly increased. (Wiesner & Pfeifer, 2014). These systems are created to utilize patients' health information, such as EHR, to provide suggestions to healthcare providers, including doctors. It is important to note that these decision-making tools are not intended to replace healthcare providers but rather to enhance their diagnostic accuracy by automatically considering all possible diseases with the aid of machine learning and AI. Recommender systems can be beneficial in numerous areas of healthcare. For instance, the machine learning algorithm behind the system could be trained using images and patterns to make suggestions for imaging results such as MRI or CT scans. Additionally, the algorithm could be fed by physicians' written reports to incorporate verbal and written information from patients. Ultimately, recommender systems are tools to facilitate decision-making for healthcare providers, regardless of the type of data they are provided.

3. METHODS

Within the scope of the MD-Advisor Project, the workflow is divided into two primary categories: data preprocessing and modeling. The project also included ontology studies to implement the NER model for future use. Once the raw data was obtained, it was pre-suitable for the use and training of the model. To be used as input, two sections related to patients' complaints are taken. The system requires the disease data to be coded using the International Classification of Diseases (ICD-10) codes as the output for diagnosis estimation. The workflow of the project is illustrated in Figure 1.



Figure 1. MD-Advisor Project Workflow

The model was selected as an RNN among deep learning models due to its recurrent structure. The deep learning model underwent training and was exported for clinical use after it successfully passed the testing and validation phases. A user interface was developed using the RNN model, based on the data from the hospital information system, to provide physicians with suggested diagnoses.

3.1. Data Source

The data set consists of three different columns. These three columns are respectively "COMPLAINTEXTITLE", "COMPLAINTEXT" and "DIAGNOSIS". The first column usually contains a summary of the complaint in a single sentence. The second column is a paragraph containing details about the patient's complaint and history. The third column contains diagnoses in the form of ICD-10 codes.

The first column is filled when the doctors enter the patient's general complaints into the system while listening to the patient. More detailed information, such as the patient's past medical history, the medications they regularly use, his current illnesses, the details of their complaint, and family history, are included in the second column. The last column consists of diagnoses made by the physician considering the patient's complaints and history.

3.2. Data Preprocessing

The raw data was obtained from the Relational Database Management System (RDBMS) of Acibadem Hospitals.



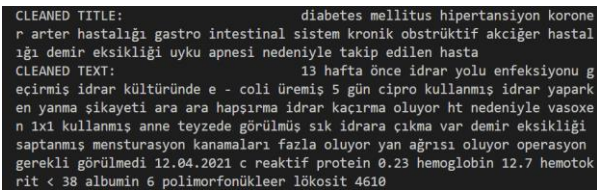
Figure 2. MD-Advisor Dataset Characteristic

Therefore, the raw data was converted to JSON format for improved efficiency and ease of processing. Subsequently, the preprocessing of the data in the JSON format was initiated. Preprocessing was essential for the modeling and could impact the overall results of the project. This was due to the presence of various issues in the raw data, such as misspellings, empty rows, and special characters. The preprocessing steps involved the elimination of empty rows, conversion of all words to lowercase,

tokenization of the dataset into individual words, and removal of special characters such as asterisks, symbols, and the 'less than' and 'greater than' symbols. Furthermore, all the abbreviations were detected and replaced with their meanings. Spell checking was also performed to correct misspelled, improperly joined, or separated words. Furthermore, the use of Regular Expressions (RegEx) was employed to locate units that had numbers before them, as these statements hold significance in the field of medicine. From the preprocessed data, specific columns that contain patients' complaints regarding their medical conditions and diseases were separated for further processing.

"Sample Complaint Title = dm htn cad gis copd patient followed for iron deficiency and sleep apnea."

"Sample Examination Notes = uti 13 wk ago, had e.coli grown in urine culture. Have used cipro for five days, complaining of burning while urinating, frequent urination, occasional sneezing, urinary incontinence, used vasoxen 1*1 because of htn. seen in mother & aunt. heavy mens. bleeding with FE deficiency gis symptoms has flank pain, operation was not considered necessary. 12.04.2021 crp 0.23, hgb 12.7, hct <38, alb 6, pmnl 4610"



CLEANED TITLE: diabetes mellitus hipertansiyon koroner arter hastalığı gastro intestinal sistem kronik obstrüktif akciğer hastalığı demir eksikliği uyku apnesi nedeniyle takip edilen hasta
CLEANED TEXT: 13 hafta önce idrar yolu enfeksiyonu geçirmiş idrar kültüründe e - coli üremiş 5 gün cipro kullanmış idrar yaparken yanma şikayeti ara ara hapsirme idrar kaçırma oluyor ht nedeniyle vasoxen 1x1 kullanmış anne teyzede görülmüş sık idrara çıkma var demir eksikliği saptanmış menstrasyon kanamaları fazla oluyor yan ağrısı oluyor operasyon gerekli görülmedi 12.04.2021 c reaktif protein 0.23 hemoglobin 12.7 hematokrit < 38 albumin 6 polimorfonükleer lökosit 4610

Figure 3. Results of the Text After Being Preprocessed

"CLEANED TITLE: Patient followed up for diabetes mellitus, hypertension, coronary artery disease, gastrointestinal system disorders, chronic obstructive pulmonary disease, iron deficiency and obstructive sleep apnea."

"CLEANED TEXT: Patient had urinary tract infection 13 weeks ago, E.coli was grown in urine culture and used ciprofloxacin for five days. Complained of burning while urinating, frequent urination and occasional urinary incontinence when sneezing. Patient used to take vasoxen once a day for hypertension. Family history reveals that hypertension is also present in her mother and aunt. Patient has heavy menstrual bleeding and iron deficiency. Pain located in the flank. Operation was not considered necessary. Blood sample analysis in 12.04.2021 show; C-reactive protein: 0.23, hemoglobin: 12.7, hematocrit < 38, albumin: 6, polymorphonuclear leukocytes 4610"

3.3. Modeling

The MD-Advisor Program followed a structured approach in the modeling stage, comprising six steps. These steps were: tokenization, label encoding, the building of Multinomial Naïve Bayes Classifiers (MultinomialNB), the building of a Support Vector Classifier (SVC), the building of an RNN model, and evaluation of the model with different techniques through training, testing, and validation.

3.3.1. Tokenization

Tokenization is widely considered as the initial step in any NLP workflow and has a significant impact on the subsequent stages. A tokenizer divides unstructured data and text into discrete elements, referred to as tokens. Since machines cannot process words directly, tokenization converts words into numerical data structures, enabling the computer or machine learning pipeline to make complex decisions or perform actions (Menzli, 2022).

The MD-Advisor Program employs tokenization techniques for generating a large dataset for training the machine learning model. The tokenization process involves applying basic techniques to separate individual words, resulting in the creation of tokens. These tokens are then transformed into word vectors, allowing for effective training of the machine learning model, as machines are unable to comprehend words in their raw form.

3.3.2. Label encoding

Label Encoding is a method imported from the Scikit Learn library for normalizing labels in a dataset. This is a function that converts categorical data into numerical values and can be compared with each other. The label encoder technique assigns numerical values to categorical labels within an interval ranging from 0 to -1 and it assigns a unique value for each label type. In the process of label encoding, if a categorical label repeats, it will consistently be assigned the same numerical value as was previously assigned.

In the MD-Advisor Program, the target labels for the machine learning models were ICD-10 codes. To make analysis and comparison easier, the Label Encoder method was used to change these codes into numerical values. This step made it possible to use these numerical labels in the machine learning process and provide a standard way to analyze and model the data.

3.3.3. Multinomial Naïve Bayes Classifiers

Naïve Bayes is a method for classifying data based on probability. It is commonly used to categorize texts, based on the analysis of the words

it contains. Unlike more complex AI-based approaches, Naïve Bayes offers a simpler solution for text classification.

The multinomial model is used for classifying non-numeric data. One of its benefits is that it is less complex than other models and can be trained with a smaller set of data, making it more efficient.

The goal of text classification is to sort the text into relevant categories. It evaluates the probability that a piece of text belongs to a particular group of similar texts. Each text is composed of multiple words that help to determine its meaning. A class is a label assigned to one or more texts that pertain to a similar subject.

The process of classifying a text involves analyzing the terms it contains and checking if they are found in other texts within the same class. This increases the likelihood that the text belongs to the same class as the previously classified texts.

The Naïve Bayes algorithm uses the Bayes theorem to categorize a text by assigning it a label. It calculates the probability of each label for a given text and assigns the class with the highest probability to that text.

The Naïve Bayes classifier is a group of algorithms that share a common assumption: that each feature being classified is independent of all other features. In other words, the presence or absence of a particular feature does not influence any other features.

Bayes theorem, named after Thomas Bayes, is a formula that calculates the probability of an event happening, taking into account prior knowledge of related conditions. It is calculated by the following formula:

where:

$P(A)$: probability of case A

$P(B)$: probability of case B

$P(B|A)$: probability of A occurring when the estimator B is given

Default parameters are the preset variables of models. These variables are usually adjusted so that the models perform best. For this reason, these values have not been changed, and the context information of the MultinomialNB Classifiers model has been chosen as the default values. These values of the parameters are as follows:

Table 1. Parameters of MultinomialNB

Parameters	Default Values
alpha	1.0
force_alpha	False
fit_prior	True

class_prior	None
-------------	------

3.3.4. Support Vector Classifier

SVC is a machine learning method used for categorizing data into different classes. It's a supervised learning approach that transforms the input data into a higher dimensional space and then locates the best dividing line between the classes, called a hyperplane.

The hyperplane is picked in a way that maximizes the gap between the closest data points of each class, which are known as support vectors. This leads to a clear distinction between the classes and makes it possible to predict new data based on these classifications.

SVC is widely used in machine learning due to its capability to manage complex, non-linearly separable data and its reliability in providing accurate classifications even with limited training data. The Scikit Learn Library provides an implementation of SVC known as Sklearn SVC.

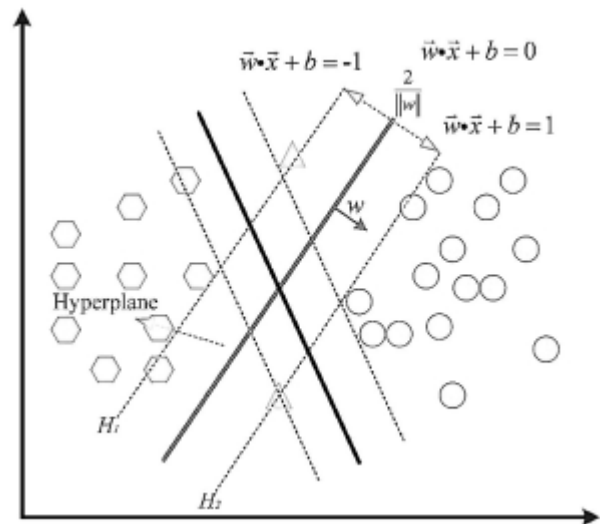


Figure 4. Illustration of the Support Vector Classifier

Since the default parameter values are determined as the best values for the performance of the model, they were not changed. Context information of the SVC model is selected as default values. These values of the parameters are as follows:

Table 2. Parameters of SVC

Parameters	Default Values
C	1.0
kernel	'rbf'
degree	3

gamma	'scale'
coef0	0.0
shrinking	True
probability	False
tol	1e-3
cache_size	200
class_weight	None
verbose	False
max_iter	-1
decision_function_shape	'ovr'
break_ties	False
random_state	None

3.3.5. Recurrent Neural Network

RNNs are a type of system that uses their previous output as an input in the next step, making them different from feedforward networks. This method allows them to have a “memory” and be useful when the input must be considered in a specific context to produce meaningful output. They are used to analyze and understand data in a specific order, such as texts, time-sensitive sensor data, and statistical data, which can not be effectively processed by feedforward networks.

In the MD-Advisor Program, the complaints indicating the patient's current state are the inputs for the model, and they are directly related to the diagnosis to be made for the patient. As a result, complaints are the most crucial factor in the diagnosis process. To accurately reflect this, the RNN model, which has a memory capability, was selected for use in the Project.

While creating the RNN model, embedding, which is a mapping method, was used. The use of embedding was chosen because it can reduce the complexity of categorical variables, allowing them to be effectively represented in the transformed space. The embedding size was chosen as 300. This helps to provide a meaningful representation of the categories.

In addition to embedding, the Long Short-Term Memory (LSTM) model was also used. LSTM is a type of RNN that has an extended memory capacity. It is commonly used for time series forecasting and serves as a building block for the layers of the RNN model. The LSTM assigns “weights” to data, allowing the RNN to effectively process new information, forget irrelevant information and give appropriate importance to data that affects the output.

The input length was chosen as 900, and the embeddings were set as trainable since there was no use of pre-trained embeddings.

In convolutional layers, Conv1D class is used to generate the output by creating a single spatial dimension from the inputs. The activation functions of these layers were selected as Rectified Linear Unit (ReLU).

A bidirectional class was used in the LSTM layer and the number of units was determined as 64.

Different length vectors temporally generated by LSTM cells were transformed into a single latent vector using the GlobalAveragePooling1D class. Then the output of the pooling layer is transmitted to the Dense layer.

The dense layer consists of 100 units and the number of units is chosen as 100.

By choosing the dropout value of 0.5, it was aimed to prevent the model from being overfit. With this method, some neurons are skipped and their weights are left unimportant.

Sparse Categorical Crossentropy is used to calculate the loss between labels and predictions. This loss function is chosen because the model will make multiclass classification and labels are encoded with Label Encoder.

Adam Optimizer, a stochastic gradient descent method, was chosen as the optimization method.

3.3.6. Model Evaluation

Model evaluation is an important process in which different statistical and mathematical methods are used to evaluate the performance and accuracy of the model. In this process, the model's predictions are compared with the actual results. The strengths and weaknesses of the classifications are determined. Based on these results, necessary changes and improvements are determined. Various criteria such as accuracy, precision, and recall are utilized to evaluate the model, depending on the particular problem and evaluation criteria.

3.3.6.1. Train/Test Split

The dataset used for the training and testing phases was obtained from AHG Hospitals. To evaluate the model correctly, 168184 rows of data were used for training. The validation set was determined as %25 of the training set. The test set was chosen as 55176 rows. Therefore, while 75.3% of the total data constituted the training and validation sets, 24.7% constituted the test set. It is important to provide the test set as well as the training data to determine whether the model is overfitting to the data set.

New patient entries are made every day in Acibadem Hospitals and added to the database. Thus, new data is generated that the model has not been tested before. With this regular data flow, the model has the opportunity to go through new testing stages.

Furthermore, since the training/test split will be insufficient, the best parameters of the model can be found by using other model evaluation techniques.

3.3.6.2. Classification Metrics

There are four classification outcomes in which the model's predictions can be placed.

- True Positives: The examined sample belongs to the class in question and its estimation is made as "belongs".
- True Negatives: The examined sample does not belong to the class in question and is estimated as "does not belong".
- False Positives: The examined sample does not belong to the class in question and its estimation is made as "belongs".
- False Negatives: The examined instance belongs to the class in question and its estimated as "does not belong".

Table 3. Confusion Matrix

	Actual True	Actual False
Predicted True	True Positives	False Positives
Predicted False	False Negatives	True Negatives

The model was evaluated with four different proportional equations created using the specified outcomes.

● Accuracy: The ratio of correct classifications to all classifications. Accuracy is a measure of how well the model is doing in its predictions in general. A high accuracy value represents that the model has mostly correctly set up parameter selections and relationships.

$$Accuracy = \frac{True\ Positives + True\ Negatives}{All\ Predictions}$$

● Precision: The ratio of true positive values to all predicted positive values. Precision can be considered as a quality determining factor. Examines how much of the predictions are relevant. When higher precision is achieved, instances are predicted in a more relevant manner.

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives}$$

● Recall: The ratio of true positive values to all actual positive values. Recall can be considered as a quantitative determinant. It examines how much of the instances return as relevant in reality. Obtaining a high recall value indicates that the most relevant instance values have been estimated.

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

● The F1 score: It is a tool used to assess the accuracy of a classification model on a specific dataset. It combines the precision and recall metrics into one single measure and is particularly useful for evaluating performance on imbalanced datasets. The F1 score is calculated as the harmonic mean of precision and recall, which involves taking the average of these two metrics.

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

The F1 score is high when both of these metrics are high and the score is low when the metrics are low. When a metric is high and the other is low, the F1 score is medium.

3.4. Ontology

An ontology is a model that defines concepts and relationships between them in a specific field. It provides a structured way to access and understand information and ensures accuracy in retrieving meaning from that information. The underlying logic behind ontologies is that in any field, the terms used are interconnected. In the medical field, for example, an ontology would connect diseases, medications, and treatment procedures. (Adelkhah, Shamsfard, & Naderian, 2019).

In the context of the MD-Advisor Program, an ontology is used to develop an NER model to assist doctors in finding relevant information about medical terms and their relationships. Having such a huge ontology in the medical field helps healthcare providers to approach diseases, medications, treatments, etc. professionally.

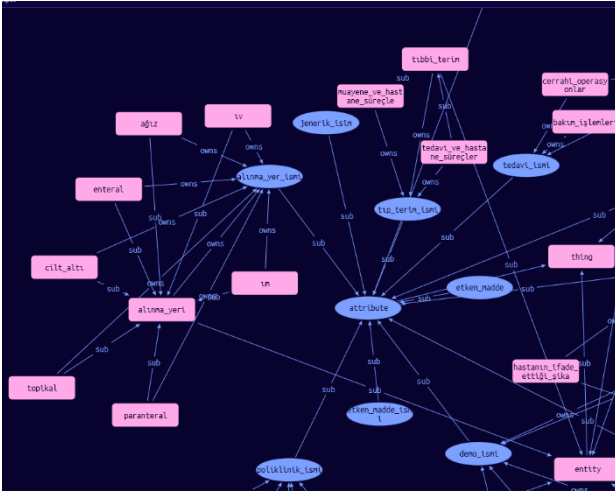


Figure 5. Ontology

3.4.1 Named Entity Recognition

Furthermore, NER is widely used in NLP to categorize words into predefined categories. (Li, 2018).

In the MD-Advisor Program, NER is used to identify the medical entities in patient notes when they are being recorded in the system. When the NER model is activated, it highlights any detected medical entities and identifies their predefined category, such as diagnosis, medication, treatment method, time expression, etc.

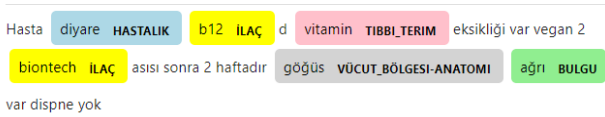


Figure 6. Named Entity Recognition Original Turkish Text

“ The patient has diarrhea DIAGNOSIS b12 DRUG vitamin MEDICINE_TERM d deficiency vegan 2 weeks after biontech DRUG vaccine has chest BODY_REGION_ANATOMY pain FINDINGS no dyspnea”

4. RESULTS

For the MD-Advisor Program, 168184 lines of internal medicine data were used as the training set for the RNN model with LSTM cells. This set contained the examination data of 167916 patients.

25% of the validation data was separated from the training set. The model was trained with 150 epochs and patient complaints were used as input. First of all, the results obtained here were examined. Accuracy was achieved as 0.8020 and the loss of accuracy after testing was 0.7500.

Owing to the regular data flow from hospitals, a test set was created with new data that the model never encountered. The test set included internal medicine data from 01.11.2022 to 01.01.2023. This

set corresponded to 32.8% of the training set and 24.7% of the total dataset.

Then, this newly created test set of 55176 lines was given to the RNN model and the outputs were examined. Accuracy 0.6145, precision 0.3163, recall 0.4460 and F1 score 0.3487 were obtained.

To compare the performance of the model and make a selection, two more models were trained and their outputs were examined, apart from the RNN model.

According to the results obtained from MultinomialNB Classifiers, accuracy was 0.2894, precision 0.0812, recall 0.1353, and F1 score 0.0916.

After the SVC model was tested with the specified test data, its accuracy value was recorded as 0.2547, precision 0.1487, recall 0.2401, and F1 Score 0.1651.

Table 4. Comparison of the Results of the Models

	RNN	Naïve Bayes	SVC
Accuracy	0.6145	0.2894	0.2547
Precision	0.3163	0.0812	0.1487
Recall	0.4460	0.1353	0.2401
F1 Score	0.3487	0.0916	0.1651

5. DISCUSSION

As in every field, applications of machine learning and deep learning models are increasing in order to create clinical decision support and prediction systems in the field of healthcare. Based on patient complaints, MD-Advisor makes diagnostic recommendations coded with ICD-10.

Diagnostic recommendations provide full code estimates. The full code estimate gives an output of five characters. The first three characters denote the category of the disease, and the last two characters give more specific details about the disease. In such a clinical classification problem, a person may have one of several disease classes or may have several diseases at the same time. For this reason, the categorization made is both multiclass and multilabel classification.

Various models and methods have been used in the MD-Advisor Program, from machine learning and deep learning algorithms to evaluation metrics.

Two machine learning and one deep learning model were built to select the best-performing model that MD-Advisor will use.

5.1. Machine Learning

Among the machine learning algorithms, MultinomialNB Classifiers and SVC models were developed.

According to the results obtained, it was observed that the performance of the SVC model was better than the MultinomialNB, except for the Accuracy value. The accuracy of the SVC model was 0.2547, lower than the MultinomialNB with 0.2894 accuracies.

The difference between precision, recall, and F1 score values of SVC and MultinomialNB models is greater than the difference between accuracy values. Therefore, the low accuracy of SVC can be ignored and it can be said that it performs better than MultinomialNB.

In prediction systems in the field of health, it is important to what extent the diseases of individuals can be predicted correctly. The Recall value gives the rate of how many of the patients with the disease are diagnosed with the specified disease. For this reason, the Recall value is among the important metrics to be considered. The fact that the Recall value of the SVC model is about two times the value of the MultinomialNB shows that the performance of the SVC is better.

5.2. Deep Learning

RNN, which contain LSTM layers, have been chosen as the deep learning algorithm. This model with memory makes it possible to perceive and process the concept of the complaints entered.

The accuracy, precision, recall, and F1 score values of the RNN model were obtained as 0.6145, 0.3163, 0.4460, and 0.3487, respectively. It has been observed that these values obtained from the RNN model are better than the values of SVC and MultinomialNB machine learning models.

The accuracy of the RNN model is about two times the accuracy of MultinomialNB and 2.5 times the accuracy of the SVC model. In clinical systems, the recall value, which shows the rate of a correct positive diagnosis, is also crucial. This value is approximately three times the recall of MultinomialNB and two times the recall of the SVC model. These results led to the selection of the RNN model for the MD-Advisor Program.

As the MD-Advisor RNN model is used by physicians, it will continue to train itself and provide better performance. These initial results from evaluation metrics will increase as usage increases and continues.

Some diagnoses entered by physicians may differ from the actual diagnosis of the patient, or the patient's complaints may be very brief. For some reports, such as military service reports, a random diagnosis can be entered. Since there is no routine control head for assessments such as check-ups, upper respiratory tract infection (J06.9, ICD-10) is usually entered. All these factors negatively affect the performance of the model. If the stated

conditions are improved, an increase in performance is also expected.

6. CONCLUSION

The MD-Advisor Program aims to enhance the accuracy and quality of diagnosis and treatment within the AHG. MD-Advisor, which is supported by AI, will assist physicians in making accurate diagnoses by addressing their needs and providing diagnostic support.

The higher precision in predictions made by the AI system among the pool of diagnoses will result in more accurate patient treatments by providing more accurate diagnoses.

MD-Advisor, which currently offers 80% accuracy on diagnosis recommendations, will provide physicians with more accurate results over time thanks to the continuous learning feature of its AI-supported systems. It is expected to effectively recognize difficult-to-detect cases, particularly in rare diseases, resulting in improved accuracy of diagnosis selection. Not only the practices of a single physician but the treatment protocols of all Acibadem physicians will be examined and presented to physicians in the form of appropriate treatment protocols. In this way, a good practice among Acibadem physicians will be developed and disseminated through MD-Advisor supported by AI.

MD-Advisor, an AI-powered clinical decision support system, has been acquired by AHG to meet its specific needs and demands, reflecting the global acceptance and utilization of such systems in healthcare. With the implementation of the MD-Advisor, there will be a dynamic process for the system to develop and become widespread, along with more demands and needs that will arise as a result of the use of AI-supported systems.

The successful use of MD-Advisor, developed by AHG's in-house resources, will be the first step for the organization to play a pioneering role in healthcare technologies and AI.

REFERENCES

- Adelkhah, R., Shamsfard, M., & Naderian, N. (2019). The ontology of natural language processing. *5th International Conference on Web Research (ICWR)*, 128-133.
- Kaur, R., Ginige, J.A., & Obst, O. (2021). A systematic literature review of automated ICD coding and classification systems using discharge summaries. *ArXiv*, 2107. 10652.
- Ma, F., Chitta, R., Zhou, J., You, Q., Sun, T., & Gao, J. (2017). Dipole: diagnosis prediction in healthcare via attention-based bidirectional recurrent neural networks. *Proceedings of the*

23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.

- Melo, M., Gusso, G.D.F., Levites, M., Massad, E., Lotufo, P.A., Zeidman, P., . . . Price, C.J. (2017). How doctors diagnose diseases and prescribe treatments: an fMRI study of diagnostic salience. *Scientific Reports*, 7(1), 1304.
- Plisson, J., Lavrač, N., & Mladenic, D. (2004). A rule based approach to word lemmatization.
- Ruder, S. (2016). An overview of gradient descent optimization algorithms. *ArXiv*, 1609.04747.
- Stark, B., Knahl, C., Aydin, M., & Elish, K. (2019). A Literature Review on Medicine Recommender Systems. *International Journal of Advanced Computer Science and Applications*, 10(8).
- Wiesner, M., & Pfeifer, D. (2014). Health Recommender Systems: Concepts, Requirements, Technical Basics, and Challenges. *International Journal of Environmental Research and Public Health*, 11(3), 2580–2607.
- Brownlee, J. (2022). Your first deep learning project in python with keras step-by-step. *Machine Learning Mastery*. Retrieved from <https://machinelearningmastery.com/tutorial-first-neural-network-python-keras/>.
- Brownlee, J. (2021). How to choose an activation function for deep learning. *Machine Learning Mastery*. Retrieved from <https://machinelearningmastery.com/choose-an-activation-function-for-deep-learning/>.
- Brownlee, J. (2019). A gentle introduction to cross-entropy for machine learning. *Machine Learning Mastery*. Retrieved from <https://machinelearningmastery.com/cross-entropy-for-machine-learning/>.
- Brownlee, J. (2017). How to visualize a deep learning neural network model in keras. *Machine Learning Mastery*. Retrieved from <https://machinelearningmastery.com/visualize-deep-learning-neural-network-model-keras/>.
- Brownlee, J. (2017). Gentle introduction to the adam optimization algorithm for deep learning. *Machine Learning Mastery*. Retrieved from <https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/>.
- Brownlee, J. (2016). 5 step life-cycle for neural network models in keras. *Machine Learning Mastery*. Retrieved from <https://machinelearningmastery.com/5-step-life-cycle-neural-network-models-keras/>.
- Brownlee, J. (2016). Multi-class classification tutorial with the keras deep learning library. *Machine Learning Mastery*. Retrieved from <https://machinelearningmastery.com/multi-class-classification-tutorial-keras-deep-learning-library/>.
- Jain, V. (2019). Everything you need to know about “activation functions” in deep learning models. *Towards Data Science*. Retrieved from <https://towardsdatascience.com/everything-you-need-to-know-about-activation-functions-in-deep-learning-models-84ba9f82c253>.
- URL-1: <https://www.dictionary.com/browse/diagnosis> [Access Date: January 2023]
- URL-2: <https://www.healthit.gov/faq/what-electronic-health-record-ehr> [Access Date: January 2023]
- URL-3: Anonymous, (2020). How Does the Gradient Descent Algorithm Work in Machine Learning? https://github.com/visionatseecs/keras-starter/blob/main/keras_intro_mlp.ipynb [Access Date: January 2023]
- URL-4: <https://www.analyticsvidhya.com/blog/2020/10/how-does-the-gradient-descent-algorithm-work-in-machine-learning/> [Access Date: January 2023]
- URL-5: https://tutorialspoint.com/deep_learning_with_keras/deep_learning_with_keras_tutorial.pdf [Access Date: January 2023]
- URL-6: <https://deeppai.org/machine-learning-glossary-and-terms/softmax-layer> [Access Date: January 2023]
- URL-7: <https://deepnotes.io/softmax-crossentropy> [Access Date: January 2023]
- URL-8: Li, S. (2018). Named Entity Recognition with NLTK and SpaCy. <https://towardsdatascience.com/named-entity-recognition-with-nltk-and-spacy-8c4a7d88e7da> [Access Date: January 2023]
- URL-9: Menzli, A. (2022). Tokenization in NLP: Types, Challenges, Examples, Tools. <https://neptune.ai/blog/tokenization-in-nlp> [Access Date: January 2023]

URL-10: Arnx, A., (2019, Jan 13). First neural network for beginners explained (with code).

<https://towardsdatascience.com/first-neural-network-for-beginners-explained-with-code-4cfd37e06eaf>

[Access Date: January 2023]

URL-11: Nielsen, M., (2019). Neural Networks and Deep Learning. Neural Networks and Deep Learning,

<http://neuralnetworksanddeeplearning.com>

[Access Date: January 2023]

URL-12: Trehan, D. (2022). Gradient Descent Explained.

<https://towardsdatascience.com/gradient-descent-explained-9b953fc0d2c>

[Access Date: January 2023]

URL-13: Srivastava, K., (2021, Jan 21). Classification – Let's understand the basics.

<https://towardsdatascience.com/classification-lets-understand-the-basics-78baa6fbff48>

[Access Date: January 2023]

URL-14: Roman, V., (2019). Supervised Learning: Basics of Classification and Main Algorithms. Towards Data Science.

<https://towardsdatascience.com/supervised-learning-basics-of-classification-and-main-algorithms-c16b06806cd3>

[Access Date: January 2023]

URL-15: Saxena, S. (2021). Binary Cross-Entropy/Log Loss for Binary Classification.

<https://www.analyticsvidhya.com/blog/2021/03/binary-cross-entropy-log-loss-for-binary-classification/>

[Access Date: January 2023]

URL-16: Godoy, D. (2018). Understanding Binary Cross-Entropy / Log Loss: A Visual Explanation.

<https://towardsdatascience.com/understanding-binary-cross-entropy-log-loss-a-visual-explanation-a3ac6025181a>

[Access Date: January 2023]

URL-17: Sharma, A., (2017). Understanding Activation Functions in Neural Networks. The Theory of Everything.

<https://medium.com/the-theory-of-everything/understanding-activation-functions-in-neural-networks-9491262884e0>

[Access Date: January 2023]

URL-18: Sharma, P. (2020). Keras Optimizers Explained with Examples for Beginners.

<https://machinelearningknowledge.ai/keras-optimizers-explained-with-examples-for-beginners/>

[Access Date: January 2023]

URL-19: Gomez, R.,(2018). Understanding Categorical Cross-Entropy Loss, Binary Cross-Entropy Loss, Softmax Loss, Logistic Loss, Focal Loss, and all those confusing names.

https://gombru.github.io/2018/05/23/cross_entropy_loss/

[Access Date: January 2023]

URL-20: Wambui, R. (2022). Cross-Entropy Loss and Its Applications in Deep Learning.

<https://neptune.ai/blog/cross-entropy-loss-and-its-applications-in-deep-learning>

[Access Date: January 2023]



Araştırma Makalesi

An Extractive Text Summarization Model for Generating Extended Abstracts of Medical Papers in Turkish

Anıl Kuş^{*1}, Çiğdem İnan Acı²

¹Toros University, Computer Technologies Department, Mersin, Türkiye

²Mersin University, Department of Computer Engineering, Mersin, Türkiye

Keywords:

Text Summarization
Extended Abstract
Medical paper
COVID-19

ABSTRACT

The rapid growth of technology has led to an increase in the amount of data available in the digital realm. This situation makes it difficult for users to find the information they are looking for within this vast dataset, making it time-consuming. To alleviate this difficulty, automatic text summarization systems have been developed as a more efficient way to access relevant information in texts compared to traditional summarization techniques. This study aims to extract extended summaries of Turkish medical papers written about COVID-19. Although scientific papers already have abstracts, more comprehensive summaries are still needed. To the best of our knowledge, automatic summarization of academic studies related to COVID-19 in the Turkish language has not been done before. A dataset was created by collecting 84 Turkish papers from DergiPark. Extended summaries of 2455 and 1708 characters were obtained using widely used extractive methods such as Term Frequency and LexRank algorithms, respectively. The performance of the text summarization model was evaluated based on Recall, Precision, and F-score criteria, and the algorithms were shown to be effective for Turkish. The results of the study showed similar accuracy rates to previous studies in the literature.

Tıp Makalelerinin Genişletilmiş Özetlerini Oluşturmak İçin Çıkarımsal Bir Türkçe Metin Özetleme Modeli

Anahtar Kelimeler:

Metin Özetleme
Genişletilmiş Özet
Tıp makalesi
COVID-19

ÖZ

Teknolojinin giderek büyümesi, dijital ortamdaki mevcut veri miktarının artmasına neden olmuştur. Bu durum, kullanıcıların bu geniş veri kümesi içinde aradıkları bilgiyi bulmalarını zorlaştırmakta ve zaman alıcı hale getirmektedir. Bu zorluğu hafifletmek için, klasik özetleme tekniklerine kıyasla daha verimli bir şekilde metinlerdeki ilgili bilgiye erişmenin bir yolu olarak otomatik metin özetleme sistemleri geliştirilmiştir. Bu çalışma, COVID-19 hakkında yazılmış Türkçe tıp makalelerinin genişletilmiş özetlerini çıkarmayı amaçlamaktadır. Bilimsel makalelerin hâli hazırda özetleri olmasına rağmen, daha kapsamlı özetlere de ihtiyaç duyulmaktadır. Türkçe dilinde COVID-19 ile ilgili akademik çalışmaların otomatik özetlemesi bildiğimiz kadarıyla daha önce yapılmamıştır. DergiPark'tan 84 adet Türkçe araştırma ve derleme makalesi alınarak bir veri kümesi oluşturulmuştur. Toplanan veri kümesinden, yaygın olarak kullanılan çıkarımsal yöntemlerden olan Terim Frekansı ve LexRank algoritmaları kullanılarak 2455 ve 1708 karakterlik genişletilmiş özetler elde edilmiştir. Metin özetleme modelinin performansı, Duyarlılık, Kesinlik ve F-skoru ölçütlerine göre değerlendirilmiş ve algoritmaların Türkçe için etkili olduğu gösterilmiştir. Çalışmanın sonuçları, literatürdeki önceki çalışmalarla benzer doğruluk oranları göstermiştir.

*Sorumlu Yazar

*(anil.kus@toros.edu.tr) ORCID ID 0000-0002-5964-3727
(caci@mersin.edu.tr) ORCID ID 0000-0002-0028-9890

e-ISSN: 2717-8579

Geliş Tarihi: 06/03/2023; Kabul Tarihi: 26/05/2023

Bilgisayar Bilimleri ve Teknolojileri Dergisi

1. INTRODUCTION

Natural Language Processing (NLP) is a subfield of Artificial Intelligence (AI) and computer science that deals with the interaction between computers and human languages (Jurafsky and Martin, 2008). The goal of NLP is to enable computers to understand, interpret, and generate human language in a way that is both natural and useful. This is a challenging task, as human language is complex, ambiguous, and context-dependent. To achieve this goal, NLP relies on various techniques from linguistics, computer science, and machine learning.

With the advancement of technology, the quantity of data in the digital area is continually expanding. As the volume of data continues to grow, it becomes increasingly difficult and tedious for users to locate the information they need. Automatic Text Summarization (ATS) systems have been designed to efficiently extract the relevant information from text documents in a more time-effective manner than manual text summarization (Bal and Gunal, 2021).

ATS has a wide range of applications, from news and information extraction to customer feedback analysis, and text-based question answering. As more and more data is generated every day, the need for ATS is becoming increasingly important, and researchers are constantly working to develop new and improved methods for generating accurate and coherent summaries (Bird, Klein and Loper, 2009)

ATS is the process of generating a shorter version of one or more documents while retaining the main idea and key information. It is one of the NLP that deals with the task of condensing a given text to a shorter version while still preserving its essential information. The goal of ATS is to generate a summary that is concise and coherent, yet still conveys the most important information from the original text.

There are two main approaches to ATS: extractive and abstractive. Extractive summarization involves identifying the most important sentences or phrases in the original text and including them in the summary. This is done by using techniques such as keyword extraction, sentence scoring, and clustering. A summary is created by selecting the sentences from the document that best represents that document in the extractive approach. In this type of summarization, the structure of the sentences remains unchanged and the positions of the words are not altered (Kemaloglu Alagöz, 2022). On the other hand, abstractive summarization involves generating new text that is not present in the original document. This is done using techniques such as text generation and natural language generation. Abstractive methods are more complex and require a deeper understanding of the content and context of the original text. But they can generate a more

human-like summary, including a paraphrase of the original text. Abstractive summarization is a complex process that involves shortening and rephrasing the original text through the use of linguistic features, making it more difficult to implement than extractive summarization. However, despite ongoing research efforts, it has not yet reached a fully mature level. (Bal and Gunal, 2021).

Both summarization approaches have their advantages and limitations. Extractive methods are generally considered to be more reliable and accurate, but they can be limited by the quality and specificity of the original text. On the other hand, abstractive methods can generate more fluent and natural-sounding summaries, but they are more challenging to implement and can be prone to errors and inaccuracies (Bird, Klein & Loper, 2009).

The research on ATS in the Turkish language commenced during the 1990s, with the study conducted by Oflazer and Kuruoz (Akulker, 2019). Turkish is among the most commonly used 20 languages in the world. The Turkish language has distinctive characteristics compared to well-studied languages in the literature, such as English, Spanish, and German (Safaya et al., 2022). It has agglutinative morphology which means that various new words can be derived by adding suffixes to a root of a word. Working with an agglutinative language such as Turkish is a real and important research issue in the context of ATS. In contrast to the other languages, there are not enough studies done on ATS in the Turkish language (Akulker, 2019). Recently, the number of studies on the Turkish language has started to increase due to its importance. One of the frequently carried out studies is the summarization of Turkish news texts. However, the studies on the summarization of academic papers are very limited. Reading scientific articles is considered to be very important in the healthcare industry and research. Successful summaries can save a significant amount of time for the reader. Although scientific papers already have abstracts, it is also necessary to have more comprehensive summaries. For researchers, it is a requirement to keep track of articles that fall within their area of interest. The challenge today is to be able to select documents that are truly relevant to their research topic from the vast amounts of information that are easily accessible within a short amount of time and to allocate sufficient time to actually read the selected documents (Celik, 2021).

The motivation of this study is to summarize scientific medical papers in Turkish about COVID-19 Pandemic using ATS methods, which have rarely worked in previous studies so far. 84 papers have been collected as the dataset from Dergipark (URL-1). The Term Frequency (TF) and LexRank algorithms, which include extractive methods and are widely used, were selected for summarization. The obtained results show an F-score of 0.52 for the TF method and 0.51 for the LexRank approach

which means that we have achieved an average level of success compared to Turkish ATS studies in the literature.

The rest of the paper is organized as follows: In Section 2, previous studies on ATS are explained. The ATS models are given in detail in Section 3. In Section 4, the results are given. Section 5 concludes the paper.

2. RELATED STUDIES

In this section, a literature review was conducted on papers published in the last seven years on ATS in Turkish is presented.

Table 1. Summary of the previous studies on extractive text summarization

Author	Year	Metric	Accuracy
Kemaloglu Alagöz	2022	BertScore+ ROUGE	0.88
Bal and Gunal	2021	Accuracy	0.78
Akulker	2019	ROUGE	0.86
Torun and Inner	2018	Human Test	0.68
Kaynar, Isik and Gormez	2017	ROUGE	0.45
Demirci, Karabudak, and Ilgen	2017	ROUGE	0.43
Hatipoglu and Omurca	2016	ROUGE	0.60

Kemaloglu Alagöz developed an ATS study that was conducted on Turkish studies in the field of computer science. In addition to pre-processing techniques prevalent in literature, a novel, format-specific pre-processing function was developed. Deep Belief Networks (DBN) were used for summarization. To assess the performance of the developed system, a customized variant of the pre-trained NLP model Bidirectional Encoder Representations from Transformers (BERT) was employed. After summarization with BERT Extractive Summarizer and DBN, the generated summaries were compared using a specialized comparison metric of BERT called BERTScore. Results showed that the system achieved an F-score of 88% in generating the summary of an article (Kemaloglu Alagöz, 2022).

Bal and Gunal developed a new extractive ATS model. In order to evaluate the efficiency of commonly utilized attributes for ATS in Turkish, sequential attribute selection methods are employed. The study was carried out using three different sets of data. The first dataset consists of 100 texts in the categories of economy, art, and sports. The second dataset consists of 20 texts from various categories. Eight different features were utilized, represented numerically using 0 and 1. These features can be described as similarity to the first sentence, similarity to the last sentence, location, length, frequency of usage, usage of question marks and exclamation marks, number of

numerical characters, and number of proper nouns. A decision tree was employed for classification, with sentences in the dataset labeled as "in summary" or "not in summary." If the majority of evaluators had decided the sentence would be a summary, the sentence was labeled as "in summary" and otherwise labeled as "not in summary." It is concluded that the accuracy of the model reached %80,84, and the presented model is considered to be efficient in ATS for the future (Bal and Gunal, 2021).

Akulker proposed an extractive ATS system that was specifically developed for the Turkish language. The system employs a statistical-based TF-IDF algorithm, as well as a hybrid approach that combines TF-IDF with the graph-based PageRank algorithm. The study primarily aims to assess the feasibility and efficacy of these algorithms for processing Turkish documents. In addition, the TF-IDF and the TF-IDF with PageRank (Hybrid) systems have been evaluated and compared against each other using ROUGE metrics during the co-selection evaluation process. According to the evaluation results, the performance of the system is dependent on both the threshold and the specific summarization algorithms employed. It was hypothesized that the precision, recall, and F-score values would improve with higher thresholds for both the TF-IDF and Hybrid systems, as human evaluators were not constrained in their selection of sentences during the summarization process, and thus, It was anticipated that summaries generated by humans would encompass a greater number of sentences than those produced using a lower threshold. The findings show that the average F-score values of the Hybrid system are better than those of the TF-IDF system, even at lower thresholds. Additionally, both systems tend to exhibit improved average F-score values with higher threshold summarization (Akulker, 2019).

Torun and Inner developed a dataset of 12,000 Turkish news articles in order to utilize the ATS system. The extraction method was employed for summarization and the texts were broken down into sentences, with abbreviations excluded from evaluation. The resulting summaries were condensed to a maximum of 5 sentences. News items that were shorter than the 5 sentences captured by the news collection tool, as well as those containing only visual content, were not considered in the evaluation. For the detection of similarity, a similar approach was proposed to the traditional methods used in ATS processes. The frequency information of words was consulted for the identification of keywords from the summarized texts (Torun and Inner, 2018).

Kaynar, Isik, and Gormez proposed a genetic algorithm-based sentence extraction method for ATS. The dataset used in the study consists of 120 Turkish news texts and their summaries. 80 documents were trained with the help of a genetic algorithm, the best weight values for the features

were determined, and then these weights were used to summarize 40 test documents, and the results were compared with the original summaries. In the study, the following steps were applied in sequence: cleaning from unnecessary words, tokenization, determining title and content, and selecting summary sentences. In this direction, 8 different features were extracted. After calculating the features of the sentences, these features were combined using a specific function to obtain the sentence score. When these scores were compared, the highest score was used in the summary sentence. Of the 120 Turkish news included in the dataset, 80 were used for training and 40 for testing. After testing, accuracy rates of 84% were reached. The system, which determined the weights through the Genetic Algorithm, has shown success with a serious accuracy rate (Kaynar, Isik, and Gormez, 2017).

Demirci, Karabudak, and Ilgen developed a summarization system for long documents in Turkish. To cluster articles based on their topics, they utilized the cosine similarity method after collecting newspaper articles dynamically from web pages via Real Simple Syndication. The Latent semantic analysis algorithm was employed for sentence scoring. To assess the performance of the system, 34 news domains were utilized, each consisting of 20, 30, 20, and 36 documents. The researchers used the ROUGE evaluation metric to compare their system's performance with manually generated summaries. The summaries were created with the assistance of 15 human evaluators. The system achieved an average recall and precision rate of 43%. The authors reported that the system's performance decreased when summarizing long papers and increased when the summarization rate was increased (Demirci, Karabudak, and Ilgen, 2017).

Hatipoglu and Omurca designed a mobile text summarization system that utilized Turkish articles from Wikipedia sources for summarization purposes. The system incorporated an Analytical Hierarchical Process (AHP) algorithm to combine structural and semantic features scores and calculate an overall score for sentences. To assess its effectiveness, the automated summaries were compared with the ones created by humans using precision and recall metrics. The study concluded that the proposed summarization method held a lot of potential in generating an understandable summary of Turkish Wikipedia articles (Hatipoglu and Omurca, 2016).

In this study, when the obtained ROUGE results are compared with the previous studies, it was observed that average success was achieved. BERT, TF-IDF, and PageRank (hybrid) algorithms were applied as an extra to the ATS models in studies that achieved more successful results in the literature.

3. MATERIAL AND METHOD

In this section, we present the details of the dataset and the algorithms utilized for ATS.

3.1 The Dataset

Since there are numerous studies, especially in the field of health, it takes a lot of time for the readers, and it is seen as a need to summarize in this field. During the COVID-19 pandemic, it was crucial to research to find a solution to the disease. Based on this, 84 papers about the pandemic which obtained from the most popular academic journals in Turkish and published on DergiPark. The journals used in the dataset are considered to be pioneers in this field such as the Journal of Istanbul Faculty of Medicine, Hacettepe University Faculty of Health Sciences Journal, Aegean Journal of Medical Sciences, and Cukurova Medical Journal.

Table 2. Paper distribution according to journals in the dataset

Journal Name	# of papers
Suleyman Demirel University Journal of Health Sciences Institute	6
Journal of Samsun Health Sciences	10
Journal Of Health Sciences	5
Health Care Academician Journal	11
Online Turkish Journal of Health Sciences	9
Mersin University Journal of Health Sciences	9
Journal of Anatolia Nursing and Health Sciences	5
Journal of Istanbul Faculty of Medicine	2
Hacettepe University Faculty of Health Sciences Journal	8
Gazi Journal of Health Sciences	4
Aegean Journal of Medical Sciences	3
Cukurova Medical Journal	12

3.2 Pre-processing

Preprocessing is a critical step that should be done after the normalization stage when summarizing the text. During the data acquisition process, it is not uncommon to encounter unwanted characters or incorrect data ordering, which can lead to unacceptable issues (Horasan and Bilen, 2020).

The sections of the dataset in this study were examined in the preprocessing according to whether they were active in the abstract or not. As a result of this review, first of all, the papers were divided into two sections. The sections (i.e.

bibliography, abstract in English, author and journal information) that will not be included in the summary have been removed from the dataset. The second one is the original abstracts (i.e. abstract in Turkish) which are used to evaluate results and compare them to the ATS's results.

The preprocessing steps used in this study are given as follows:

Data cleaning: For NLP applications, certain characters such as numeric characters, punctuation marks, etc. are generally considered non-essential and may be removed or ignored during the preprocessing stage. However, directly removing them might not be enough for the data-cleaning step (Karayigit, Aci and Akdagli, 2021).

Tokenization: This step involves dividing a text corpus or sentence into smaller elements, such as words, phrases, or n-grams. The dataset became more manageable, modifiable, and analyzable. It is significant for enhancing the precision of NLP modeling and analysis by making the linguistic and semantic structures of the data more discernible.

Stop-words removal: Stop-words removal is a common preprocessing step in NLP that involves removing high-frequency, function words from a text corpus. In the Turkish language, these words, such as "ve", "veya", "fakat", "yani" etc., are considered irrelevant for certain NLP tasks. In our study, Natural Language Toolkit (NLTK) was used to remove all stop-words.

Normalization: All texts were converted to lowercase, converted words to their root form and removed characters such as whitespaces, short lines, etc. which are not useful.

3.3 Term Frequency (TF)

TF is one of the main concepts of the extractive summarization method. In this method, firstly we need separate all the words according to their roots and put them in tables where they occur throughout the text. The second step is to calculate all word frequencies by eliminating the stopwords. Then, word frequency scores will be divided by maximum frequency which represents the total score of the paper. The result shows the rate of the words. Some keywords such as "COVID-19", "pandemics" or "Corona" are multiplied by 5 due to an increase in word frequency and selection of these words. After that, sentence scores are calculated by determining sentence lengths. And the words which have high scores will be included in the sentences of the summarization part. In the experiments, it was seen that a summary of a maximum of 30 sentences had the highest ROUGE value.

$$TF(t) = (\text{Number of times term } t \text{ appears in a document}) / (\text{Total number of terms in the document}) \quad (1)$$

Upon examination of the most successful summary generated using the TF method, it can be

seen that the original paper consists of 33,369 characters, while the summary generated by the system is 2,460 characters long and the original abstract is 2,493 characters long. Therefore, it can be said that the summary generated using the TF method is almost as long as the original abstract. The F-score of the summarized text was found to as 0.52.

In the study, the studies published on coronavirus are summarized and the ten publications with the highest ROUGE value are as in table 3.

Table 3. Precision, Recall, and F-score of summaries that have top 10 scores using the TF method.

#	Recall	Precision	F-score
1	0.46	0.61	0.52
2	0.46	0.44	0.45
3	0.40	0.49	0.44
4	0.51	0.38	0.44
5	0.55	0.36	0.44
6	0.40	0.47	0.43
7	0.53	0.35	0.42
8	0.63	0.31	0.42
9	0.36	0.48	0.41
10	0.49	0.34	0.40

3.4 LexRank

LexRank is a graph-based algorithm for ATS. The algorithm creates a graph structure to represent the semantic similarity among sentences in the text and then performs a PageRank analysis on the graph to identify the most important sentences or paragraphs in the text. The PageRank algorithm is commonly used to measure the importance of a website in search engine results. By using the LexRank algorithm, the summary generated will be more meaningful and accurate, as it takes into account the semantic similarity between sentences. The algorithm is widely used for ATS and generally produces good results.

Mathematically, the LexRank algorithm works by representing sentences in a text as nodes in a graph and then using similarity measures such as cosine similarity to determine the edges between the nodes. Cosine similarity is the fundamental measure utilized for evaluating content similarity. It is widely preferred as one of the most common methods for determining the similarity between two texts by comparing them (Celik, 2021). The similarity measure between two sentences is calculated as the dot product of their vector representations. Once the graph is constructed, the LexRank algorithm uses the PageRank algorithm, which is based on a random walk model, to assign a score to each sentence. The PageRank algorithm

works by iteratively updating the score of each sentence in the graph based on the scores of the sentences that are linked to it. The final scores assigned to each sentence by the algorithm represent the importance of that sentence in the text. The highest-scored sentences are then selected to create the summary of the text.

In this study, the LexRank algorithm was applied with sentence weighting, and the parameter for sorting the highest-scoring sentences was set to a minimum of 10. A commonly used threshold value of 0.1-0.2 was chosen to determine the similarity measure, which has led to variations in obtaining broader summaries. When weighing the words based on their frequency, specifically those considered as keywords, they were multiplied by a coefficient of 2 or 3 depending on the context.

The original text consists of 24,492 characters, while the system generated a summary of 962 characters in summaries generated by the LexRank algorithm. The original summary length in this study is 1,567 characters. The performance of the generated summary was calculated as 0.51 F-score.

In Table 4, the ROUGE results of the summaries of the papers are given with the LexRank algorithm.

Table 4. Precision, recall, and F-score of summaries that have top 10 scores using the LexRank algorithm.

#	Recall	Precision	F-score
1	0.43	0.62	0.51
2	0.47	0.52	0.49
3	0.38	0.54	0.45
4	0.41	0.46	0.43
5	0.42	0.42	0.42
6	0.44	0.41	0.42
7	0.53	0.35	0.42
8	0.37	0.48	0.42
9	0.38	0.44	0.41
10	0.38	0.43	0.40

3.5 Design and Implementation

This section provides a detailed account of the technical specifications involved in the development process of the ATS model, along with an explanation of the working principles of the model which utilizes the extracting method with TF and LexRank systems. The subsequent sections will elaborate on each module developed for the summarization process, detailing each step comprehensively.

The ATS model was created by leveraging the Python programming language in the Anaconda Integrated Development Environment, utilizing the Spyder Framework. The hardware setup that was

employed throughout the development phase comprised an Intel Core i5-7200 CPU with a processor speed of 2.50 GHZ, 8192 MB of RAM, and 500 GB of Hard Disk space.

Figure 1 illustrates a comprehensive overview of the ATS model architecture that we developed along with a step-by-step demonstration.

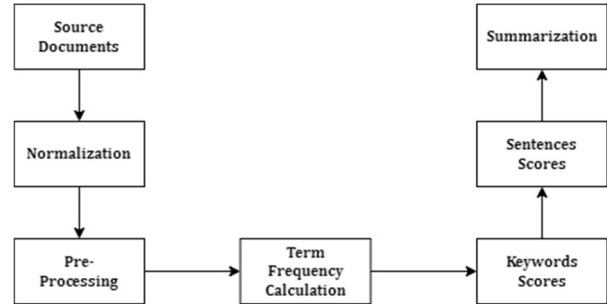


Fig.1. The architecture of the ATS model with TF and keyword scores

4 RESULTS AND DISCUSSIONS

Evaluating the performance of ATS systems is a crucial step in the development and improvement of these systems. The evaluation of ATS systems is a challenging task as it is difficult to quantify the quality of a summary, as it is a subjective task.

Different evaluation measures can be applied to assess the performance of ATS systems. Some of the most commonly used evaluation measures include Recall-Oriented Understudy for Gisting Evaluation (ROUGE), Metric for Evaluation of Translation with Explicit Ordering (METEOR), and Consensus-Based Image Description Evaluation (CIDEr). It includes measures to automatically determine the quality of a summary by comparing it to other (ideal) summaries created by humans. It is observed that the evaluation using ROUGE is the most popular in studies conducted in this field. In our study, the ROUGE value was also used for summary evaluations.

To assess the performance of our ATS study, we utilized the most frequently employed evaluation metrics in the field, which are recall, precision, and the F-score derived from these values.

The four components of the confusion matrix serve as the foundation for evaluating the performance of a classifier. True Positive (TP), False Negative (FN), False Positive (FP), and True Negative (TN) (Karayigit, Aci and Akdagli, 2021). TP refers to the number of sentences in the generated summary that are also present in the reference summary. FN refers to the number of sentences in the reference summary that are not present in the generated summary. FP refers to the number of sentences in the generated summary that are not present in the reference summary. TN refers to the number of sentences in the original text that are not

present in either the reference summary or the generated summary.

Based on these definitions, precision, recall, and F-score can be calculated using the following formulas:

$$\text{Precision (P)} = \text{TP} / (\text{TP} + \text{FP}) \quad (2)$$

$$\text{Recall (RC)} = \text{TP} / (\text{TP} + \text{FN}) \quad (3)$$

$$\text{F-score} = 2 * (\text{P} * \text{RC}) / (\text{P} + \text{RC}) \quad (4)$$

Regarding the TF method and LexRank, we can see that the success of the summarization scores is similar, but LexRank provides shorter summaries. Figure 2 shows the most successful summaries acquired with these methods.

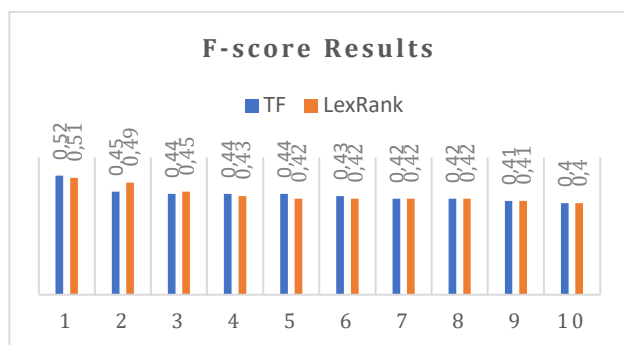


Fig.2. F-score results of two methods

When comparing the two methods, the lengths of the top 10 summaries with the highest F-score are as follows:

- The TF method produces summaries with an average length of 2,455 characters.
- The LexRank algorithm produces summaries with an average length of 1,708 characters.

Based on this result, it can be said that LexRank achieves the same level of performance with shorter summaries. However, the shortness of the summaries does not necessarily mean that they are semantically successful. This comparison is based on the similarity to the original text.

The ranking or frequency values in the BOW (Bag-of-Words) representations obtained from the two methods have shown similarity. The reason for the differences is attributed to the fact that the TF method operates on word-level weights, while the LexRank method operates on sentence-level weights. However, both approaches share similarities in terms of applying normalization and pre-processing steps, as well as extracting summaries based on frequency.

The reason for the similar performance of the summaries obtained using LexRank and TF methods is believed to be due to several factors. Firstly, both methods are based on word frequency, making them frequency-based approaches. Secondly, both LexRank and TF methods fall into the category of extractive summarization techniques. Instead of generating new sentences, they extract important information directly from the original text. Furthermore, both LexRank and TF

methods prioritize the original text content. They aim to capture and highlight important information present in the original text.

5 CONCLUSION

This paper presents the performance analysis of the Turkish ATS system that applies two different methods. Increasing keywords scores with the TF method is more successful and its results are promising. To the best of current academic knowledge, there is no study available in the literature that systematically collates and summarizes the existing papers on the subject of COVID-19. In the field of health, it is widely recognized that a comprehensive summary of existing studies would be extremely valuable for professionals working in this field, as it would provide them with a comprehensive understanding of the current state of knowledge on the subject and aid them in their work.

In this study, a system was designed to automatically collect and summarize texts written about the coronavirus. The system was used to summarize 84 different papers. Similarity detection was performed on the summarized texts. The similarity between the original abstracts and the summaries generated by the system was calculated using the ROUGE value.

According to the results, it was observed that the summaries generated by the system were similar to the original texts. Furthermore, the length of the summaries in terms of character count was also analyzed. It was determined that in the TF method, as the length of the summary increased, the success of the summary also increased. In contrast, in the LexRank method, shorter summaries were found to be more successful compared to the TF method.

In the respect to the experiments, we can explain that the most frequent words and keywords do not always reflect the subject of the document correctly. It has been observed that the titles or keywords increase the results and reflect the documents more properly. In addition, since there is not enough study on Turkish ATS, the dataset will contribute to researchers who study the Turkish language.

In future work, we plan to apply other summarization methods, especially abstractive methods using machine learning and deep learning algorithms. We believe that in order to reach more reliable results preprocessing steps must be done carefully. To ensure the effectiveness of summarized texts, they should be compared to a more dependable standard. Although human evaluation may require more resources, it is widely considered to provide a more accurate assessment than comparing the summarized text to the original abstract.

REFERENCES

- Akulker, E. (2019). Extractive Text Summarization For Turkish Using Tf-Idf And Pagerank Algorithms (Doctoral dissertation). *The Graduate School Of Natural And Applied Sciences Of Atılım University, Turkey.*
- Bal, S. and Sora Gunal, E. (2021). A New Model On Automatic Text Summarization For Turkish. *Eskisehir Technical University Journal Of Science And Technology A- Applied Sciences And Engineering*, 22(2), 189-198.
- Bird, S., Klein, E., & Loper, E. (2009). Natural language processing with python. O'Reilly Media.
- Celik Ozkan, A. E. (2021). Structured Abstract Extraction System for Turkish Academic Publications (Doctoral dissertation). *Hacettepe University, Turkey.*
- Demirci F., Karabudak, E. and Ilgen, B. (2017). Multi-Document Summarization for Turkish News. *International Artificial Intelligence and Data Processing Symposium (IDAP)*, pp. 1-5.
- Hatipoglu, A. and Omurca, S.I. (2016). A Turkish Wikipedia Text Summarization System for Mobile Devices. *IJ. Information Technology and Computer Science*, vol.1, pp. 1-10.
- Horasan, F. And Bilen, B. (2020). Extractive Text Summarization Systems For News Texts. *International Journal Of Applied Mathematics Electronics and Computers*, 8(4), 179-184.
- Jurafsky, D., and Martin, J. H. (2008). Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition (2nd ed.). Prentice Hall.
- Karayigit, H., İnan Aci, C. and Akdagli, A. (2021). Detecting abusive Instagram comments in Turkish using convolutional Neural network and machine learning methods. *Expert Systems with Applications*, 174(March).
- Kaynar, O., Emre Isik, Y. and Gormez, Y. (2017). Genetic Algorithm Based Sentence Extraction for Automatic Text Summarization. *Journal of Management Information Systems*, 3 (2) , 62-75.
- Kemaloglu Alagöz, N. (2022). Automatic Text Summarization With Deep Learning (Doctoral dissertation). *Suleyman Demirel University, Turkey.*
- Safaya, A., Kurtulus, E., Goktogan, A. and Yuret, D. (2022). Mukayese: Turkish NLP Strikes Back, 846-863.
- Torun, H. and Inner, A. B. (2018). Detecting similar news by summarizing Turkish news. *26th IEEE Signal Processing and Communications Applications Conference, SIU 2018*, 1-4.
- URL-1:
<https://dergipark.org.tr/tr/pub/page/about>
[last accessed: 2023/02/06]