T.C.
KAHRAMANMARAŞ SÜTÇÜ İMAM ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

İSTATİSTİKSEL DOĞAL DİL İŞLEMEDE DERİN ÖĞRENME YÖNTEMLERİ KULLANILARAK ÇEVRİMİÇİ TÜRKÇE AKADEMİK DERLEM ÇÖZÜMLENMESİ

BARIŞ BABÜROĞLU

YÜKSEK LİSANS TEZİ

ENFORMATİK ANABİLİM DALI

KAHRAMANMARAŞ 2019

T.C.
KAHRAMANMARAŞ SÜTÇÜ İMAM ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

İSTATİSTİKSEL DOĞAL DİL İŞLEMEDE DERİN ÖĞRENME YÖNTEMLERİ KULLANILARAK ÇEVRİMİÇİ TÜRKÇE AKADEMİK DERLEM ÇÖZÜMLENMESİ

BARIŞ BABÜROĞLU

Bu tez,
Enformatik Anabilim Dalında
YÜKSEK LİSANS
derecesi için hazırlanmıştır.

KAHRAMANMARAŞ 2019

Kahramanmaraş Sütçü İmam Üniversitesi Fen Bilimleri Enstitüsü öğrencisi Barış BABÜROĞLU tarafından hazırlanan "İSTATİSTİKSEL DOĞAL DİL İŞLEMEDE DERİN ÖĞRENME YÖNTEMLERİ KULLANILARAK ÇEVRİMİÇİ TÜRKÇE AKADEMİK DERLEM ÇÖZÜMLENMESİ" adlı bu tez, jürimiz tarafından 25/06/2019 tarihinde oy birliği ile Enformatik Anabilim Dalında Yüksek Lisans tezi olarak kabul edilmiştir.

Prof. Dr. Mehmet TEKEREK (DANIŞMAN) …….
Bilgisayar ve Öğretim Teknolojileri Eğitimi
Kahramanmaraş Sütçü İmam Üniversitesi

Dr. Öğr. Üyesi Hasan BADEM (ÜYE) …….
Bilgisayar Mühendisliği.
Kahramanmaraş Sütçü İmam Üniversitesi

Dr. Öğr. Üyesi Özge KELLECİ (ÜYE) …….
Eğitim Programları ve Öğretimi ABD
Hasan Kalyoncu Üniversitesi

Yukarıdaki imzaların adı geçen öğretim üyelerine ait olduğunu onaylarım.

Prof. Dr. Mustafa YAZICI …….

Fen Bilimleri Enstitüsü Müdürü

TEZ BİLDİRİMİ

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, ayrıca tez yazım kurallarına uygun olarak hazırlanan bu çalışmada, alıntı yapılan her türlü kaynağa eksiksiz atıf yapıldığını bildiririm.

Barış BABÜROĞLU

Bu çalışma Kahramanmaraş Sütçü İmam Üniversitesi, Bilimsel Araştırma Projeleri Koordinasyon Birimi tarafından desteklenmiştir.
Proje No: 2017/4-20 YLS

Not: Bu tezde kullanılan özgün ve başka kaynaktan yapılan bildirişlerin, çizelge, Şekil ve fotoğrafların kaynak gösterilmeden kullanımı, 5846 sayılı Fikir ve Sanat Eserleri Kanunundaki hükümlere tabidir.

İSTATİSTİKSEL DOĞAL DİL İŞLEMEDE DERİN ÖĞRENME YÖNTEMLERİ KULLANILARAK ÇEVRİMİÇİ TÜRKÇE AKADEMİK DERLEM ÇÖZÜMLENMESİ

BARIŞ BABÜROĞLU

ÖZET

Doğal dil, insanları diğer canlılardan ayıran ve insanların iletişim kurmasını sağlayan en temel özelliklerden biridir. Dil, insanın duygu ve düşüncelerini ifade etmede kullandığı ve kültürlerin nesiller boyunca aktarılmasını sağlayan bir araçtır. Günlük hayatta karşılaşılan yazılar ve sesler birer doğal dil örneğidir. Doğal dilde birçok kelime zamanla yok olurken diğer taraftan yeni kelimeler de türetilmektedir. Bu yüzden doğal dil işleme (DDİ) süreci insan için bile karmaşık yapıya sahipken, bilgisayar ortamında işlenmesi de zor olmaktadır. İnsanların dili nasıl kullandığını dil bilim alanı incelemektedir. Dil bilimciler ve bilgisayar bilimcilerinin ortak çalışmasını gerektiren doğal dil işleme çalışmaları, insan bilgisayar etkileşiminde önemli rol oynamaktadır. Doğal dil işleme çalışmaları, yapay zekâ teknolojilerinin, dil bilimi alanında kullanılması ile artmıştır. Yapay zekâ çalışma alanlarından olan derin öğrenme yöntemleri ile doğal dile yakın seviyede platformlar geliştirilmektedir. Dili anlama, makine çevirisi ve sözcük etiketleme için geliştirilen platformlar derin öğrenme yöntemlerinden faydalanmaktadır. Derin öğrenme mimarilerinden olan tekrarlayan sinir ağları (Recurrent Neural Network - RNN), metin veya ses verileri gibi sıralı verileri işlemede tercih edilmektedir. Bu çalışmada bir RNN türü olan iki yönlü uzun-kısa vadeli bellek (Bidirectional Long Short - Term Memory - BLSTM) kullanılarak Türkçe sözcük etiketleme modeli önerilmiştir. Önerilen sözcük etiketleme modeli, doğal dil araştırmacılarına, kendi analizlerini gerçekleştirme ve kullanabilme imkânı verecek bir platform ile sunulmaktadır. İki yönlü LSTM kullanılarak geliştirilen platformun geliştirilme aşamasında uzman görüşü ile geri bildirimler alınarak, sözcük etiketleyicinin hata oranı azaltılmıştır.

Anahtar Kelimeler: Doğal Dil İşleme, Derin Öğrenme, İki Yönlü Uzun-Kısa Vadeli Bellek, Özyinelemeli Sinir Ağları, Sözcük Etiketleme

Kahramanmaraş Sütçü İmam Üniversitesi

Fen Bilimleri Enstitüsü

Enformatik Anabilim Dalı, Haziran / 2019

Danışman: Prof. Dr. Mehmet TEKEREK

Sayfa sayısı: 62

ANALYSING TURKISH ACADEMICAL CORPUS USING DEEP LEARNİNG METHODS IN STATISTICAL NATURAL LANGUAGE
(M.Sc. THESIS)

BARIŞ BABÜROĞLU

ABSTRACT

Natural language is one of the most fundamental features that distinguish people from other living things and enable people to communicate each other. Language is a tool that enables people to express their feelings and thoughts and to transfers cultures through generations. Texts and audio are examples of natural language in daily life. In the natural language, many words disappear in time, on the other hand new words are derived. Therefore, while the process of natural language processing (NLP) is complex even for human, it is difficult to process in computer system. The area of linguistics examines how people use language. NLP, which requires the collaboration of linguists and computer scientists, plays an important role in human computer interaction. Studies in NLP have increased with the use of artificial intelligence technologies in the field of linguistics. With the deep learning methods which are one of the artificial intelligence study areas, platforms close to natural language are being developed. Developed platforms for language comprehension, machine translation and part of speech (POS) tagging benefit from deep learning methods. Recurrent Neural Network (RNN), one of the deep learning architectures, is preferred for processing sequential data such as text or audio data. In this study, Turkish POS tagging model has been proposed by using Bidirectional Long-Short Term Memory (BLSTM) which is an RNN type. The proposed POS tagging model is provided to natural language researchers with a platform that allows them to perform and use their own analysis. In the development phase of the platform developed by using BLSTM, the error rate of the POS tagger has been reduced by taking feedback with expert opinion.

Key words: Natural Language Processing, Deep Learning, Bidirectional Long-Short Term Memory, Recursive Neural Networks, POS Tagging

Kahramanmaraş Sütçü İmam University

Graduate School of Science and Technology

Department of Informatics, June / 2019

Supervisor: Prof. Dr. Mehmet TEKEREK

Page Numbers: 62

TEŞEKKÜR

Tez çalışmam süresince engin bilgi ve tecrübelerinden faydalandığım, desteğini eksik etmeyen ve anlayışlı tavrıyla uygun bir çalışma ortamı sağlayan değerli hocam Prof. Dr. Mehmet TEKEREK'e teşekkürlerimi sunarım.

Son olarak, hayatımın her alanında olduğu gibi bu tez çalışmasında da sonsuz destekleri ile sürekli yanımda hissettiğim aileme; üniversite yıllarından itibaren motivasyonumu yitirdiğim anlarda bile desteğini benden esirgemeyen eşim Ece BABÜROĞLU'na çok teşekkür ederim.

İÇİNDEKİLER

ŞEKİLLER DİZİNİ

ÇİZELGELER DİZİNİ

Sayfa No

SİMGELER VE KISALTMALAR DİZİNİ

DDİ : Doğal Dil İşleme

BNC : İngiliz Ulusal Derlemi

STD : Sözlü Türk Derlemi

ICLE : Uluslararası Öğrenci İngilizcesi Derlemi

TICLE : Türkçe Uluslararası Öğrenci İngilizcesi Derlemi

COCA : Çağdaş Amerikan İngilizcesi Derlemi

TUD : Türkiye Ulusal Derlemi

SVM : Destek vektör makinaları

YSA : Yapay Sinir Ağı

NER : Adlandırılmış Varlık Tanıma

SRL : Semantik Rol Etiketleme

CNN : Konvolüsyonel Sinir Ağı

RNN : Tekrarlayan Sinir Ağı

BRNN : İki Yönlü Tekrarlayan Sinir Ağı

LSTM : Uzun-Kısa Vadeli Bellek

BLSTM : İki Yönlü Uzun-Kısa Vadeli Bellek

BLSTM-RNN: İki Yönlü Uzun Kısa Süreli Bellek Tekrarlayan Sinir Ağı

1. GİRİŞ

Doğal dil, insanların kendilerini ifade etmeleri ve iletişim kurabilmeleri için kullanılan bir araçtır. Chomsky, dilin çocukluk yıllarında duyulandan, doğal bir dile dönüşümünün, insanın genetik yapısıyla ilişkili olduğunu ifade etmektedir (Chomsky, 1986). İnsanların dili nasıl edindiği, ürettiği ve anladığı dil biliminin araştırma alanıdır. Dil bilimciler dilsel ifadeleri işlemek için kural tabanlı yaklaşımlar öne sürmüşlerdir. Fakat doğal dil kullanımının üretkenliği kurallara uymayabilmektedir. Bu yüzden doğal dil ifadelerine istatistiksel yaklaşımlar uygulanmıştır. İstatistiksel yaklaşımlar ile doğal dil için ortak ilkeler oluşturulmaya çalışılmıştır. İstatistiksel dil modelleri, dil bilimi ve bilgisayar biliminin alt bilim dalı olan doğal dil işleme (DDİ) çalışmalarında başarı ile uygulanmıştır (Schütze & Manning, 1999). DDİ'nin amacı, doğal dilleri bilgisayarlar tarafından oluşturma ve anlamadaki sorunları incelemektir (Young, Hazarika, Poria, & Cambria, 2018). DDİ insanlar tarafından üretilen sesleri ve metinleri işleyerek insan bilgisayar etkileşiminin sağlanmasına yardımcı olmaktadır.

DDİ insan dilinin otomatik analizi ve gösterimi için teorik olarak motive edilmiş hesaplama teknikleridir (Cambria & White, 2014). Her yaşta insanın sosyal medyaya ulaşabildiği bir ortamda, üretilen veri miktarı, her geçen gün artarak devam etmektedir. İnsanlar tarafından doğal olarak oluşturulan veriler, doğrudan işlenecek durumda değildir. Bu yüzden insan makine iletişimini sağlamak için verileri anlamlandırma ve verimli kullanabilme çabası birçok alanın birlikte çalışmasını gerektirmiştir. İlk zamanlarda yapılan çoğu DDİ çalışmaları, tek tek sözcüklere odaklanmışken 19. yüzyılın sonlarına doğru sözcüklerin birbirleriyle olan ilişkisine ve bütün üzerinde anlam bilim çalışmalarına yönelmiştir (Cambria & White, 2014).

DDİ çalışmaları ilk olarak metni anlamak için ses veya metinden özellik çıkarımı yapan bir ön işlemeden geçmektedir. Ardından Şekilbilim (morphological), sözdizim (syntactic), anlambilim (semantic) ve söylev (discourse) işleme çalışmaları gerçekleştirilebilmektedir. Bu çalışma alanları sözcük kökleri, sözcük bağlamları ve anlambilim açısından bazı zorluklara sahiptir. Zorlukları aşmak için geliştirilen dilbilgisine dayalı kural tabanlı DDİ çalışmaları (Brill, 1992), (Gimenez & Marquez, 2004), el yapımı özelliklere dayanmaktadır. El yapımı özellikler zaman almakta ve yetersiz kalmaktadır (Young, Hazarika, Poria, & Cambria, 2018). Eksikliklerin giderilmesi için geliştirilen yapay zekâ yöntemlerinin önemli bir uygulaması olan derin öğrenme, yapay sinir ağı

yapısı, güçlü donanımı ve büyük veri girdisi ile daha iyi sonuçlar elde edilmesini sağlamıştır (Song & Lee, 2013).

1965 yılında derin öğrenmenin temeli kabul edilen çok katmanlı bir perceptron türü algoritma (Ivakhnenko & Lapa, 1965) önerilmiştir. Fakat o yıllardaki basit bir ağın eğitiminin uzun sürmesi ve yüksek hesaplama maliyetlerinden dolayı, destek vektör makinaları gibi el ile hazırlanmış özelliklere sahip modeller (Cortes & Vapnik, 1995) kabul görmüştür (Şeker, Dirib, & Balık, 2017). Yakın zamanda grafik işleme birimi (GPU) ve diğer donanımsal gelişmeler sayesinde hesaplama maliyetleri düştüğünden, çok sayıda gizli katmandan oluşan yapay sinir ağları tekrar kullanılmaya başlanmıştır (Schmidhuber, 2015). Bu doğrultuda DDİ çalışmalarında, makine çevirisi, bilgi alma, metin özetleme, soru cevaplama, bilgi çıkarma, konu modelleme ve sözcük etiketleme gibi görevlere, derin öğrenme uygulanmasına odaklanılmaktadır (Şeker, Dirib, & Balık, 2017), (Young, Hazarika, Poria, & Cambria, 2018).

Basit bir derin öğrenme çerçevesinin, adlandırılmış varlık tanıma, anlamsal rol etiketleme ve sözcük etiketleme gibi birçok DDİ görevinde, en modern yaklaşımlardan daha iyi performans gösterdiği ortaya konulmuştur (Collobert, ve diğerleri, 2011). DDİ alanında istatistiksel yöntemler, kural tabanlı yöntemlerden daha başarılı olmaktadır. Bu alanda sözcük etiketlemesi, sözcük türlerinin sözcüklere atanması ile gerçekleştirilmektedir (sözcük/isim, sıfat, fiil, vb.). Etiketler bilgisayarların cümlede ifade edileni anlamasında kolaylık sağlamaktadır. Ancak sözcükler farklı bağlamlarda kullanıldığı zaman farklı anlamlar ifade edebilmektedir. Örneğin ‘yüz’ sözcüğü kullanıldığı bağlama göre isim veya fiil etiketini alabilmektedir. Verilen örneği incelediğimizde, her sözcüğü etiketlemenin söz konusu olmadığı görülmektedir. Sözcük etiketlemede belirsizliği gidermek için etiketlenecek sözcüğün öncesinde kullanılan sözcüklere (geri yönde) ve sonrasında kullanılan sözcüklere (ileri yönde) bakılarak sözcük, doğru etiket sınıfına eklenebilmektedir. Etiketleme işlemlerinde, iki yönde ki bilgileri kullanarak olasılık üreten, bir Tekrarlayan Sinir Ağı (RNN) türü olan, iki yönlü uzun-kısa vadeli bellek (BLSTM) ağlarının, sıralı verileri etiketlemek için doğal dil işleme çalışmalarında çok etkili olduğu gösterilmektedir (Wang, Qian, K. Soong, He, & Zhao, 2015). BLSTM mimarisi, dil modelleme (Sundermeyer, Schlüter, & Ney, 2012), (Sundermeyer, Ney, & Schluter, 2015), dili anlama (Yao, Zweig, Hwang, Shi, & Yu, 2013), makine çevirisi (Sundermeyer, Alkhouli, Wuebker, & Ney, 2014) ve sözcük etiketlemesi gibi doğal dil işleme alanındaki

uygulamalar için üstün performans elde edilmesine yardımcı olmaktadır (Wang, Qian, K. Soong, He, & Zhao, 2015).

1.1 Doğal Dil İşleme

Doğal dil işleme (DDİ), Türkçe veya İngilizce gibi insan dillerinin bir bilgisayar tarafından kullanılmasıdır. Bilgisayar programları genellikle basit programlar tarafından verimli ve net bir Şekilde ayrıştırma sağlamak için tasarlanmış özel dilleri kullanır. Daha doğal olarak oluşan diller belirsizdir ve resmi tanımlara uymayabilirler. Birçok DDİ uygulaması, doğal bir dilde cümle, kelime veya karakter dizileri üzerinde olasılık dağılımı tanımlayan dil modellerine dayanmaktadır.

Etkili bir DDİ oluşturmak için genellikle sıralı verileri işlemek için uzmanlaşmış teknikler kullanılmaktadır. Çoğu durumda, doğal dili tek tek karakter dizisi yerine, sözcük dizisi olarak işleme tercih edilmektedir. Toplam olası kelimelerin sayısı çok büyük olduğu için, kelime tabanlı dil modelleri son derece yüksek boyutlu ve seyrek bir ayrık alanda çalışılmaktadır. Böyle bir alanın modellerini hem hesaplamalı hem de istatistiksel anlamda verimli hale getirmek için çeşitli stratejiler geliştirilmiştir (Goodfellow, Bengio, & Courville, 2016).

1.2 Derlem

Derlem İngilizce literatürde corpus olarak geçmekte olup, Latince beden anlamına gelmektedir. Sözcügün çoğul biçimi olarak literatürde corpora kullanılmaktadır. Derlem, boyutça büyük metin koleksiyonlarının elektronik biçimlerde bilgisayar ortamında tutulan dilsel kesitleri olarak tanımlanabilir. Bir derlem içindeki metinlerin türleri, yazarı, yayın tarihi ve yeri vb. hakkında bilgi verir (Baker, Hardie, & McEnery, 2006). Ayrıca derlemlerin büyük metin arşivlerinden farkı bu metinlerin sesbilimsel, sözdizimsel ve anlambilimsel olarak bir standart çerçevesinde işaretlenmiş olmasıdır. Bir derlemin dengeli bir Şekilde bir dili temsil etmesi için mümkün olduğunca çok sayıda metin kategorisini kapsaması gerekmektedir. Bir derlemin dengesini bilgili ve sezgisel kararlar dışında değerlendirmek için somut bir yöntem yoktur. Araştırma alanlarına ve kapsamlarına göre derlemlerin türü belirlenmektedir. Derlemler genel ve özel olmak üzere iki ana kategoride ele alınmaktadırlar. Derlemler kendi içinde alana özgü derlemler, referans derlemleri, çok dilli derlemler, paralel, öğrenci derlemleri, art zamanlı ve monitör (çok büyük boyutlu daha

çok güncel dili kapsayan derlem) olmak üzere türlere ayrılmaktadır (Aksan & Aksan, 2018).

1.2.1 Genel Derlem

Genel derlemler sözlü veya yazılı metinleri ayrı ayrı veya ikisini bir arada bulundurabilir. Amaç, farklı alanlardan ve türlerden gelen metinleri dengeli olarak temsil etmektir. Örneğin British National Corpus (BNC), modern İngilizcenin 100 milyon kelimelik ve 4048 yazılı metin ve on milyon kelimelik kopyalanmış sözlü veri içeren genel bir derlemidir. Ulusal ve bölgesel gazete ve dergilerden örneklenen metinleri, popüler ve akademik kitapları, üniversite denemelerini, e-posta örneklerini, yayınlanmamış mektupları ve farklı yaşlardan, kurumlardan ve okurlardan gelen raporları içerdiğinden, dengeli ve temsili bir modern İngilizceyi içermektedir. BNC 'nin genel derlem olarak başarısı diğer diller için de temel tasarım ilkelerinin benimsenmesini sağlamıştır (Aksan & Aksan, 2018).

1.2.2 Özel Derlem

Özel derlemler küçük boyutlu ve tür veya alan bakımından özelleşmiş derlemlerdir. Bu tür derlemler daha çeşitlidir ve daha fazla sayıda bulunurlar. Profesyonel ve akademik alanlarda özelleşmiş derlem oluşumu daha çok gözlenmektedir (Aksan & Aksan, 2018).

1.2.3 Yazılı Derlem

Yazılı derlemin ve modern zamanlarda İngilizcenin ilk derlemi The Brown Derlemidir. Derlem verilerini oluşturan metinler 15 kategoriden örneklenen yazılı medyadan toplanmaktadır (Aksan & Aksan, 2018).

1.2.3 Sözlü Derlem

Genel veya yazılı derleme kıyasla, bir dilin sözlü derlemini oluşturmak ve açıklama eklemek zordur. Kayıt teknolojilerindeki gelişmeler nedeniyle sözlü derlemlerin sayısında bir artış olmuştur. Örneğin, Türkçe için mevcut ve dilbilimsel olarak güvenilir konuşma derlemi, Sözlü Türk Derlemi (STD) geliştirilmiştir (Ruhi, ve diğerleri, 2010).

1.2.4 Tek Dilli Derlem

Tek dilli derlem en sık görülen derlem türüdür. Sadece bir dilde metinler içerir. Derlem genellikle konuşmanın bölümleri için etiketlenir ve geniş bir kullanıcı yelpazesi tarafından kullanılabilir. Örneğin bir kelimenin doğru kullanımını kontrol etmek veya en doğal kelime kombinasyonlarını aramak gibi bilimsel görevler gibi çeşitli görevler için kullanılmaktadır (Corpus Types, 2019).

1.2.5 Paralel Derlem

Bir paralel derlem iki dilli derlemlerden oluşur. Bir derlem diğerinin çevirisidir. Örneğin, bir roman ve onun çevirisi paralel bir derlem oluşturmak için kullanılabilir. Her iki dilin de hizalanması gerekir, yani karşılık gelen bölümlerin, genellikle cümle veya paragrafların eşleştirilmesi gerekir. Kullanıcı daha sonra bir dilde bir kelime veya cümlenin tüm örneklerini arayabilir ve sonuçlar diğer dilde karşılık gelen cümlelerle birlikte görüntülenir (Corpus Types, 2019).

1.2.6 Çok Dilli Derlem

Çok dilli bir derlem, hepsi aynı metnin çevirileri olan ve paralel derlemler ile aynı hizada olan birkaç dilde metinler içerir. Kullanıcının ikiden fazla hizalı derlem seçmesini sağlar ve arama aynı anda tüm dillerdeki çeviriyi gösterir. Yalnızca iki dil seçildiğinde, çok dilli bir derlem paralel bir derlem gibi davranır. Kullanıcı ayrıca tek dilli bir derlem olarak kullanmak için bir dille çalışmaya karar verebilir (Corpus Types, 2019).

1.2.7 Öğrenen Derlem

Öğrenen bir derlem, bir dil öğrenenlerin kullandığı, yazılı metinleri veya konuşulan dilin kodlarını içeren bir derlemdir. Bu derlem yabancı dil öğrenirken öğrencilerin hatalarını ve sorunlarını incelemek için kullanılmaktadır (Corpus Types, 2019). Örneğin, Uluslararası Öğrenci İngilizcesi Derlemi (ICLE) (Granger, 2003) ve onun alt derlemi olan Türkçe Uluslararası Öğrenci İngilizcesi Derlemi (TICLE) (Kilimci & Can, 2009), son yıllarda bağlamların öğretilmesinde bir araştırma kaynağı olmuştur (Aksan & Aksan, 2018).

1.2.8 Art Zamanlı Derlem

Bir artzamanlı derlem, farklı dönemlerden metinler içeren bir derlemdir ve dil gelişimi veya değişimini incelemek için kullanılır. Ses kayıt teknolojilerinin yakın tarihine bakıldığında, Helsinki Diachronic İngilizce Metinleri Derlemi (Rissanen, ve diğerleri, 1991) gibi artzamanlı derlem zaman içinde yazılı dili temsil eder (Aksan & Aksan, 2018).

1.2.9 Monitör Derlem

Bunlara ek olarak bir monitör derlem, sesli veya görsel araçlar ile geliştirilmiş metinleri içerir. Monitör derlemde sunulan diğerlerinden farklıdır; yeni bilginin eklenmesiyle sürekli büyümektedir. Çağdaş Amerikan İngilizcesi Derlemi (COCA) (Davies, 2009) bu tür İngilizce için iyi bilinen bir derlemdir (Aksan & Aksan, 2018).

1.3 Derlem ve Doğal Dil İşleme

Dilbilimsel çalışmalar, derlem tabanlı ve derlem odaklı analizlerin ilerlemesiyle hız kazanmıştır. Depolama kapasitelerinin artması ve mevcut yazılımın karmaşıklığı gibi bilgisayar teknolojisindeki gelişmeler, derlem dilbiliminin bugünkü statüsünü elde etmesini sağlamıştır. Halen, derlem dilbilimi, veri toplama ve analiz etmede sofistike bir metodolojiye dönüşmektedir. Türk dilbiliminde derlemler oluşturmak ve kullanmak, DDİ çalışmalarıyla motive olmuş yeni bir girişimdir. Bu geç katılımın en önemli sebeplerinden biri, 2000'li yılların başına kadar hiçbir kurumsal destek bulunmamasıdır. Derlem yapısı, kurumsal olarak desteklenmesi gereken emek ve zaman isteyen bir faaliyettir. Türkiye'deki az sayıda dilbilim bölümü ve bu tür çalışmaların takdir edilmemesi ve finanse edilmemesi, bu tür kurumların eksikliğine neden olmaktadır. Türkçe'nin diğer dillere göre toplanmış ve iyi belgelendirilmiş farklı temsil seviyelerinde dilbilgisi ve genel Türkçe açıklamaları sınırlı kalmaktadır.

Dilbilimsel analiz için bilinen ilk elektronik derlem, Köksal (1976) tarafından "otomatik morfolojik analiz" için yapıldı. Bu çalışmada, günlük gazetelerden rastgele seçilen 1534 kelimelik bir metin örneğinden oluşan bir derlem üzerinde testler ve değerlendirmeler gerçekleştirilmiştir. Bu çalışmada Türkçenin zengin morfolojisini ve olasılıksal kombinasyonlarını tanımak ve ayrıca potansiyel dil alanları için daha büyük derlemler oluşturmak için potansiyel uygulama alanlarına dikkat çekilmektedir.

Modern Türkçeyi temsil etmek üzere tasarlanan ve derlenen ilk elektronik derlem, Orta Doğu Teknik Üniversitesi Türkçe Derlem'dir. Bu Türkçe derleminin geliştiricileri temel amacın hem betimleyici hem de teorik çalışmalarda faydalı olacağını ümit ederek, Türkçe üzerinde dengeli bir yazılı derlem tasarlamak olduğunu belirtmektedir (Say & Özge, 2004).

2000 'li yıllarda Türkçenin geniş çaplı bir genel referans derlemine duyulan ihtiyaç daha belirgin hale gelmiştir. Bu zorlukla başa çıkabilmek için, Mersin Üniversitesi'ndeki bir grup dilbilimci, Türkçe'nin referans birliğini oluşturmaya karar vermiştir. Son ürün, büyük ölçekli (50 milyon kelime) genel amaçlı bir çağdaş Türk derlem kaynağının ücretsiz kaynağı olan Türkiye Ulusal Derlem'i (TUD). TUD'nin yapımındaki tasarım ilkeleri, BNC yapısında küçük değişikliklerle uyarlanmıştır. STD, dilbilimsel analizler için kullanılabilen türünün tek örneğidir. Sözlü derlem araştırmacılarının karşılaştıkları kaynak yaratma, bakım ve yayma konusunda farklı zorlukların çoğunu aşan sözlü derlem alanında öncü bir çalışmadır (Aksan & Aksan, 2018).

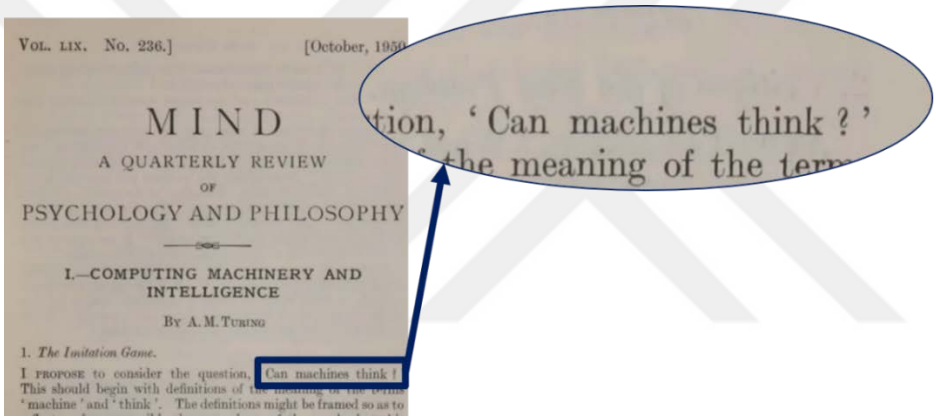
Bahsedilen DDİ araştırmacılarının çalışmaları ile aslında Türkçe için bir dilbilimsel analiz dönemi başlatılmıştır. Mevcut Türkçe çalışmalarındaki çoğu dilsel eser, söylem analizi, pragmatik veya sözdizimi gibi az sayıda alana odaklanmaktadır. Nadiren semantik ya da sözlükbilim üzerine ya da başka alanlarda, zenginleştirilmiş veri setlerine ihtiyaç duydukları için eserler bulunmaktadır. Bu nedenle, Türkçenin iyi dengelenmiş ve temsili bir derlemi, dilin birikmiş ve belgelenmiş potansiyellerinin ve onun temsilci veri setlerinin göreceli olarak küçük olduğu bir dili çalışmak için bir zorunluluktur (Aksan & Aksan, 2018).

21. yüzyılın başlarında donanımsal ve yazılımsal eksikliklerden dolayı yavaş ilerlemekte olan DDİ çalışmaları yapay zekâ alanındaki gelişmeler ile hız kazanmıştır (Cambria & White, 2014).

2. YAPAY ZEKA

İnsanlar uzun zamanlar düşünen makineler yaratmayı hayal etmişlerdir. Programlanabilir bir bilgisayarın ilk kez tasarlanmasından yüz yıl önce zeki olup olmayacağını merak etmişlerdir. Bugün, yapay zekâ pratik uygulamalar ve aktif araştırma konuları ile gelişen bir alandır. Rutin emeğin otomatikleştirilmesi, konuşma veya görüntülerin anlaşılması, tıpta teşhis konulması ve temel bilimsel araştırmaların desteklenmesi için akıllı yazılımlar geliştirilmektedir (Goodfellow, Bengio, & Courville, 2016).

1950'de Alan Turing, düşünen makineler yaratma olasılığı hakkında spekülasyon yaptığı bir dönüm noktası makale yayınladı (Şekil 1) (Turing, 1950).


VOL. LIX. No. 236.] [October, 1950

MIND

A QUARTERLY REVIEW

OF

PSYCHOLOGY AND PHILOSOPHY

I.—COMPUTING MACHINERY AND INTELLIGENCE

BY A. M. TURING

1. *The Imitation Game.*

I PROPOSE to consider the question, Can machines think ?

tion, ‘Can machines think ?’

the meaning of the term

Şekil 1 1950 Mind Dergisinde A.M. Turing tarafından yazılan makale

Amerikalı bilgisayar bilimci olan Jhon McCharty ilk Yapay zekâ terimini ve Lisp programlama dilini icat etmiş ve sembolik yapay zekâ araştırmalarına önemli katkılarda bulunmuştur. John McCarthy, 1955'te Dartmouth Koleji'nde bir Yaz Araştırması Projesi yürütmüştür. Bir grup profesör ve öğrenci, makinelerin normalde insanlar için ayrılmış problemleri çözmek için dili, soyutlamaları ve kavramları nasıl kullandıklarını bulmaya çalıştıklarını önermiştir. Yapay zekâ alanı, Dartmouth Koleji'ndeki bir konferansta, John McCarthy ile kurulmuştur (McCarthy, Minsky, Rochester, & Shannon, 1955). O yıllarda yapay zekâ araştırmacıları tarafından kullanılan bir bilgisayar Şekil 2 ‘de gösterilmektedir.

Şekil 2 IBM 702: Birinci nesil yapay zekâ araştırmacıları tarafından kullanılan bir bilgisayar

Yapay zekanın tarihi fantezilerin, olasılıkların, gösterilerin ve vaatlerin tarihidir. Homer, akşam yemeğinde tanrıları bekleyen mekanik "tripodlar" eserini yazdığından beri, hayal edilen mekanik asistanlar kültürümüzün bir parçası olmuştur. Bununla birlikte, yalnızca son yarım yüzyılda, AI topluluğu, düşünce ve akıllı davranış mekanizmaları hakkında hipotezleri test eden ve böylece daha önce sadece teorik olanaklar olarak var olan mekanizmaları gösteren deneysel makineler üretmiştir (Buchanan, 2006). Birçok uygulama mevcut durumda geliştirilmiştir ve neredeyse sınırsız sayıda gelecek uygulama için potansiyel mevcuttur.

Yapay zekanın temel bilimsel amacı, doğal veya yapay sistemlerde akıllı davranışı mümkün kılan ilkeleri anlamaktır. Araştırmacılar bunu aşağıdaki adımlar ile uygular;

• Doğal ve yapay maddelerin analizi,

• Akıllı uygulamalar oluşturmak için gerekenler hakkında hipotezler oluşturmak ve test etmek,

• Görevleri gerçekleştiren hesaplamalı sistemler tasarlamak, inşa etmek ve denemek.

Amaç, tek başına akıllı düşünmeyi gerçekleştirmek değildir. Akıllı düşünme, sadece daha akıllı davranışa yol açtığı sürece önemli olmaktadır. Yapay zekanın ana mühendislik amacı, kullanışlı, akıllı eserlerin tasarımı ve sentezidir (Poole & Mackworth, 2019).

İnsanlık tarihi boyunca insanlar kendilerini modellemek için teknolojiyi kullanmışlardır. Her yeni teknolojiden akıllı uygulamalar veya akıl modelleri oluşturmak için faydalanılmaya çalışılmıştır. Saat mekanizması, hidrolik, telefon anahtarlama sistemleri, hologramlar, analog bilgisayarlar ve dijital bilgisayarların tümü hem akıl için teknolojik metaforlar hem de akıl modelleme mekanizmaları olarak önerilmiştir (Poole & Mackworth, 2019).

16. YY'dan beri insanlar düşünce ve aklın doğası hakkında yazmaya başlamışlardır. Haugeland (1985) tarafından "Yapay zekanın Büyükbabası" olarak tarif edilen İngiliz felsefeci Hobbes, düşüncenin yüksek sesle konuşmak veya kalem ve kâğıtla cevap yazmak gibi sembolik bir mantık olduğu fikrini ortaya atmıştır. Sembolik düşünce fikri, Fransız matematikçiler Descartes, Pascal ve zihin felsefesinde öncü olan diğerleri tarafından daha da geliştirilmiştir (Poole & Mackworth, 2019).

Bilgisayarların gelişimi ile sembolik işlemler fikri daha da somut hale gelmiştir. İngiliz matematikçi Babbage, ilk genel amaçlı bilgisayar olan Analitik Motoru tasarlamıştır. Yirminci yüzyılın başlarında, hesaplamanın anlaşılması için yapılan çok çalışma ve öneri vardı. Gerçek bilgisayarlar bir kez kurulduktan sonra, ilk bilgisayar uygulamalarından bazıları yapay zekâ programlarıydı. Örneğin, Samuel (1959) bir dama programı oluşturdu ve dama oynamayı öğrenen bir program uygulamıştır. Bu program, 1961'de Connecticut eyalet dama şampiyonunu yenmiştir. Wang (1960), Principia Mathematica'daki (Russell & Whitehead, 1910) her mantık teoremini (yaklaşık 400) ispatlayan bir program uygulamıştır.

Üst düzey sembolik akıl yürütme çalışmalarına ek olarak, nöronların çalışma şeklinden ilham alan düşük seviyeli öğrenme konusunda da çok çalışma bulunmaktaydı. McCulloch ve Pitts (1943), basit bir eşik "biçimsel nöron" un Turing-complete bir makinenin temeli olabileceğini göstermiştir. Bu sinir ağları için ilk öğrenme Minsky (1952) tarafından tanımlanmıştır. İlk önemli çalışmalardan biri Rosenblatt'ın perceptron algoritmasıdır (Rosenblatt, 1958). Sinir ağları üzerindeki çalışmalar, 1968 tarihli Minsky ve Papert kitabının (1988) kitabında öğrenilen temsillerin akıllı düşünme için yetersiz olduğu görüşünün savunulması ile düşüşe geçmiştir.

1960'larda ve 1970'lerde, sınırlı etki alanları için doğal dil anlama sistemleri geliştirilmiştir. Örneğin, Daniel Bobrow'un (1967) STUDENT programı, doğal dilde ifade

edilen lise cebir görevlerini çözebilmekteydi. Winograd'ın (1972) SHRDLU sistemi, sınırlı bir doğal dil kullanarak, simüle edilmiş bir blok dünyadaki görevleri tartışıp yürütebilmeyi başarabilmişti. CHAT-80 (Warren & Pereira , 1982), kendisine verilen ülkeler, nehirler ve diğer coğrafi sorular hakkındaki gerçekleri veri tabanına dayanarak doğal dilde cevaplayabilmekteydi. Bu sistemlerin tümü, sınırlı kelime hazinesi ve cümle yapısı kullanarak yalnızca sınırlı alanlarda mantıklı olabilmekteydi. 2012 yılında ilginç bir şekilde, TV programı Jeopardy!'de dünya şampiyonu yenen IBM'in Watson'ı, CHAT-80 ile benzer bir teknik kullanmıştır (Lally, ve diğerleri, 2012).

1990'lı ve 2000'li yıllarda yapay zekanın alt disiplinlerinde algı, olasılık ve karar-teorik akıl yürütme, planlama, somutlaştırılmış sistemler, makine öğrenmesi ve diğer birçok alanda büyük gelişmeler yaşanmıştır (Poole & Mackworth, 2019).

Son yapay zekâ araştırmaları yeni derin öğrenme yöntemlerinin kullanımına odaklanmaktadır. Onlarca yıl problem çözümleri makine öğrenmesi yöntemleri ile oluşturulmuş modellere (örneğin, destek vektör makinaları ve lojistik regresyon) dayanmaktaydı (Young, Hazarika, Poria, & Cambria, 2018).

2.1. Makine Öğrenmesi

Makine Öğrenmesi matematiksel ve istatistiksel işlemler ile veriler üzerinden çıkarımlar yaparak tahminlerde bulunan sistemlerin bilgisayarlar ile modellenmesidir. Makine öğrenme teknolojisi, modern toplumun birçok yönünü güçlendirir: web aramalarından sosyal ağlarda içerik filtrelemeye, e-ticaret web sitelerindeki önerilere kadar kameralar ve akıllı telefonlar gibi tüketici ürünlerinde giderek daha fazla bulunmaktadır. Makine öğrenme sistemleri, görüntüdeki nesneleri tanımlamak, konuşmayı metne dönüştürmek, haber öğelerini, yayınları veya ürünleri kullanıcıların ilgileriyle eşleştirmek için kullanılmaktadır (LeCun, Bengio, & Hinton, 2015). Yani makine öğrenmesi, verilen verilerden defalarca tekrar ederek öğrenme işlemidir. Makine öğrenmesi yaklaşımlarının kalitesi doğru özelliklerin seçimine bağlıdır (Blum & Langley, 1997).

2.1.1 Veri Önişleme

Makine öğrenmesinde farklı algoritmalar farklı veri türleri üzerinde çalışmayı tercih edebilmektedir. Bu yüzden algoritmaların verimli çalışabilmesi için veriyi hazır hale getirilmesi gerekmektedir. Birçok uygulama alanı, karmaşık ön işleme gerektirir çünkü

orijinal girdi, birçok makine öğrenme mimarisi için çok zor olan bir formda gelmektedir (Goodfellow, Bengio, & Courville, 2016).

2.1.2 Eksik Veriler

Veri setlerinde eksik veriler olabilmektedir. Bazı algoritmalar eksik veriler ile çalışamamaktadır.

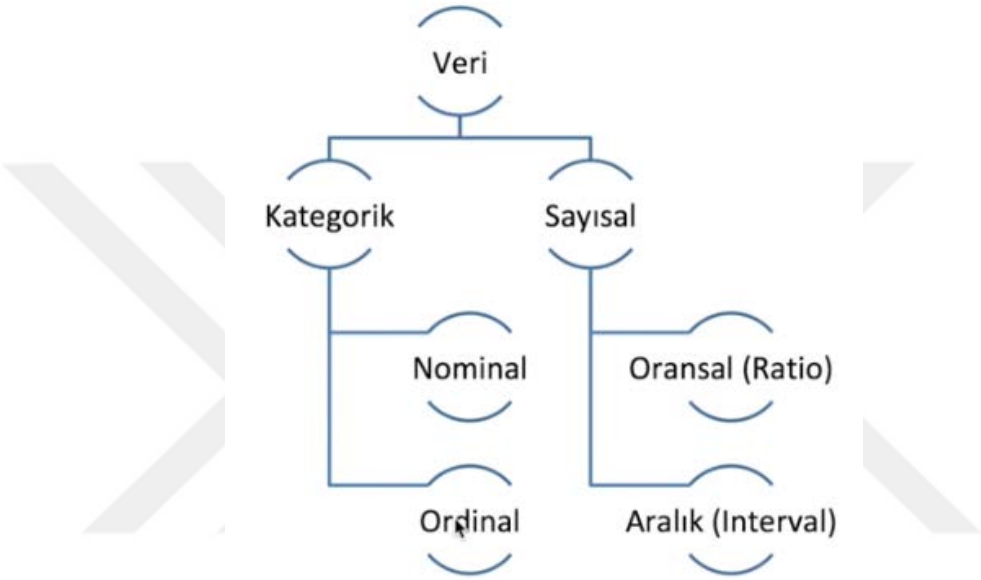
Şekil 3 Yaş kolonunda eksik veri bulunan örnek veri seti

Yukarıdaki veri setinde yaş kolonunda eksik veriler olduğu görülmektedir. Sayısal eksik verileri farklı Şekilde düzeltme yöntemleri bulunmaktadır. Eksik veriler için yeni değerlerler yazdırılabilmektedir. Hatalı okunan veya bozulmanın sebep olduğu eksik verilerin yerine rastgele değer atanabileceği gibi genelde en kolay yöntem olarak ilgili kolondaki değerlerin ortalaması alınarak (mean imputation) eksik veriler düzeltilebilmektedir.

Şekil 4 Eksik veri yerine ortalama değer ataması (mean imputation)

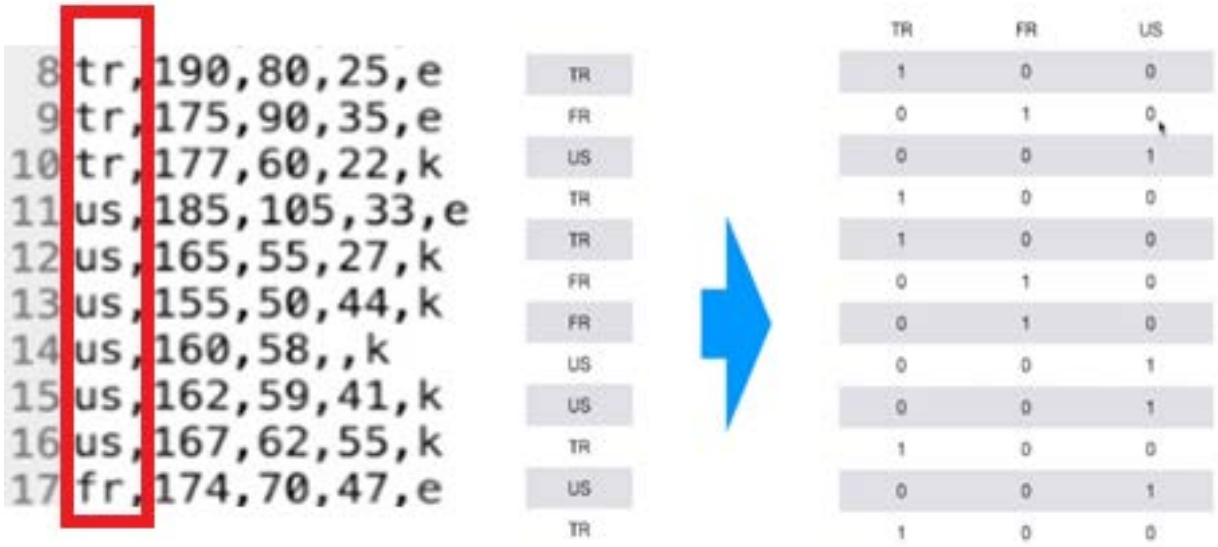
2.1.3 Veri Türü Dönüşümü

Veriler kategorik ve sayısal olarak iki ana gruba ayırılmaktadır (Şekil 5). Kategorik verilerin altında ordinal veriler olarak bahsedilen veriler aralarında büyüktür/küçüktür ilişkisi kurulabilen ama ölçülemeyen verilerdir. Nominal değerler sıralanamayan ve ölçülemeyen verilerdir. Sayısal veriler altında oransal veriler birbirlerine göre orantılanabilen (çarpılıp/ bölünebilen) verilerdir. Aralık verileri ise çarpma ve bölmeyi kabul etmeyen verilerdir.



Şekil 5 Veri Türleri

Makine öğrenmesi algoritmaları kullanılırken bu veri türleri arasında geçiş yapılabilmesi gerekebilir. Örneğin aşağıdaki veri setinde boy, kilo ve yaşı kullanarak cinsiyet tahmini yapmak istediğimizde bu değerler ülkelere göre farklılık gösterebilme ülke kolonunu tr;1, us;2, fr;3 şeklinde sayısal değere dönüştürmek istediğimizde Amerika Türkiye'nin 2 katıdır veya Amerika + Türkiye = Fransa gibi bir ilişki kurulamaz. Bu Şekilde Polinomial verileri sayısal değere dönüştürmek için her etiket kolon başlığına taşınabilir (Şekil 6).



Şekil 6 Polinomial Veri Dönüşümü

2.1.4 Veri Birleştirme

Verilerdeki problemleri düzeltip veriyi hazır hale getirdikten sonra parça parça düzenlenen bu verileri birleştirerek tek bir veri seti haline getirmemiz gerekmektedir. Eksik verileri düzeltildikten ve veri türü dönüşümü gerçekleştirildikten sonra veriler birleştirilerek Şekil 7 'deki veri seti elde edilir.

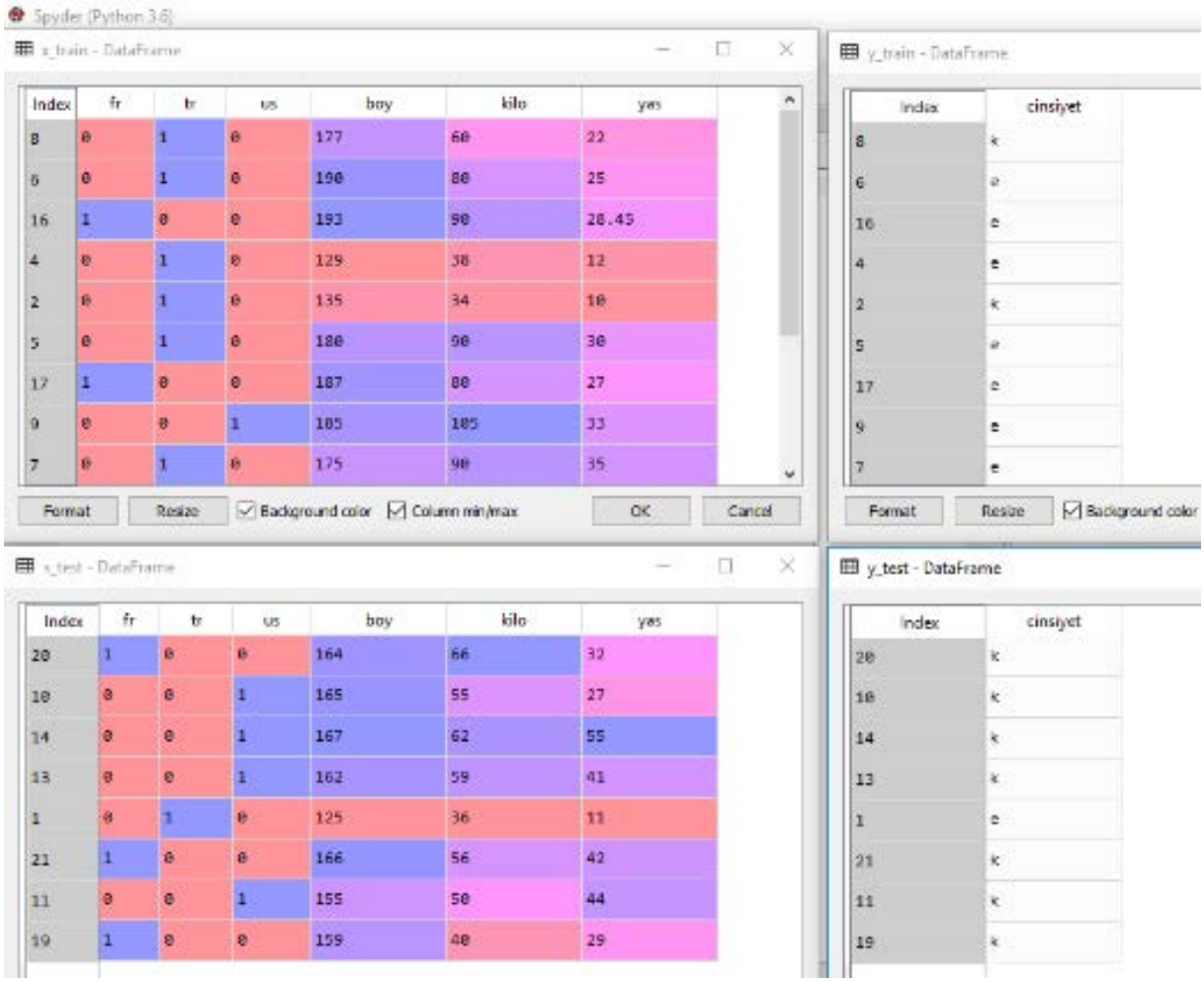
	fr	tr	us	boy	kilo	yas	cinsiyet
0	0.0	1.0	0.0	130.0	30.0	10.00	e
1	0.0	1.0	0.0	125.0	36.0	11.00	e
2	0.0	1.0	0.0	135.0	34.0	10.00	k
3	0.0	1.0	0.0	133.0	30.0	9.00	k
4	0.0	1.0	0.0	129.0	38.0	12.00	e
5	0.0	1.0	0.0	180.0	90.0	30.00	e
6	0.0	1.0	0.0	190.0	80.0	25.00	e
7	0.0	1.0	0.0	175.0	90.0	35.00	e
8	0.0	1.0	0.0	177.0	60.0	22.00	k
9	0.0	0.0	1.0	185.0	105.0	33.00	e
10	0.0	0.0	1.0	165.0	55.0	27.00	k
11	0.0	0.0	1.0	155.0	50.0	44.00	k

Şekil 7 İşlenmeye hazır birleştirilmiş veri seti

2.1.5 Test ve Eğitim Kümesi Oluşturma

Makine Öğrenme algoritmasının veri üzerinden öğrenmeyi sağlaması ve algoritmanın başarısının ölçülebilmesi için eğitim ve test kümesi olarak ikiye bölünmesi

gerekmektedir. Genel olarak ideal sonuçlar sağlayan parçalama kuralı kabul edilen Percentage Split kullanılarak veri kümesinin 2/3 ü eğitim için 1/3 ü test için kullanılmaktadır (Dobbin & Simon, 2011). Bu kural kullanılarak oluşturulan Şekil 8 'de gösterilen eğitim kümesinde verilen cinsiyetlerden yola çıkarak test kümemizdeki boy, kilo ve yaş niteliklerine göre tahminde bulunması istenilmektedir (Kümeler verilerden rastgele oluşturulmaktadır).



Şekil 8 Eğitim ve Test Kümeleri

2.1.6 Öznitelik Ölçekleme

Veriler üzerinde makine öğrenmesi algoritması kullanırken özniteliklerin veri istatistiksel özellikleri (ortalama, dağılım, standart sapma vb) birbirinden farklı olmaması gerekmektedir. Veri bilimi için verilerin farklı ölçeklerden gelmesi bir problem olarak karşımıza çıkmaktadır. Bu yüzden bu veriler ortak bir ölçekte birleştirilmelidir. Veri biliminde iki klasik yöntem kullanılmaktadır; Standartlaştırma ve Normalleştirme.

Normalleştirme hesabı verileri belli bir aralığa indirgemeyi (0 ile 1 arası vb) amaçlarken, standartlaşma hesabı verilerin birbirleriyle olan farkını bozmadan orta değerden ne kadar uzaklaştığını hesaplamayı amaçlar. Veri kümemizde marjinal değer var ise normalleştirmede problem yaşanırken standartlaştırma bu durumdan daha az etkilenir. Bu yüzden genelde standartlaştırma yöntemi tercih edilmektedir (Şekil 9).

	0	1	2	3	4	5
0	-0.632456	0.866025	-0.408248	0.450494	-0.296579	-0.247171
1	-0.632456	0.866025	-0.408248	1.00825	0.509655	0.0341619
2	1.58114	-1.1547	-0.408248	1.13696	0.912772	0.357695
3	-0.632456	0.866025	-0.408248	-1.60891	-1.18344	-1.18495
4	-0.632456	0.866025	-0.408248	-1.35148	-1.34468	-1.3725
5	-0.632456	0.866025	-0.408248	0.579207	0.912772	0.503051
6	1.58114	-1.1547	-0.408248	0.879537	0.509655	0.221717
7	-0.632456	-1.1547	2.44949	0.793728	1.51745	0.784384
8	-0.632456	0.866025	-0.408248	0.364686	0.912772	0.971939
9	1.58114	-1.1547	-0.408248	0.70792	0.832148	0.315495
10	-0.632456	0.866025	-0.408248	-1.43729	-1.50593	-1.46628
11	-0.632456	0.866025	-0.408248	-1.566	-1.50593	-1.3725
12	1.58114	-1.1547	-0.408248	0.321782	0.106538	2.09727
13	-0.632456	-1.1547	2.44949	-0.278878	-0.377202	0.357695

Şekil 9 Veri Kümesi üzerinde uygulanan standartlaştırma

2.2 Makine Öğrenmesi Algoritmaları

En çok kullanılan problem tipleri tahmin ve sınıflandırma problemleridir. Kategorik verilerde bir tahmin yapılmak isteniyorsa Sınıflandırma (Classification) problemi, sayısal verilerde bir tahmin yapılmak isteniyorsa Tahmin (Prediction) problemi olarak adlandırılmaktadır.

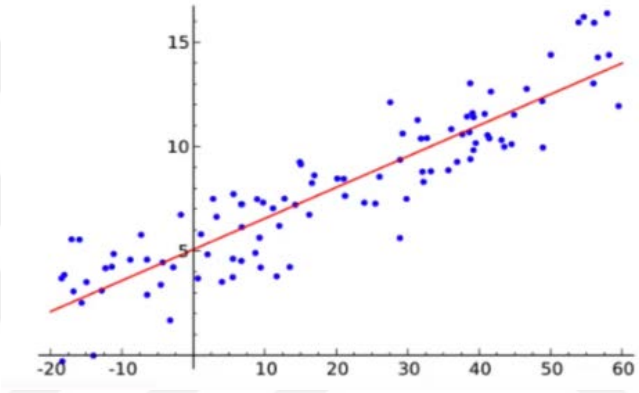
2.2.1 Basit Doğrusal Regresyon (Simple Linear Regression)

Verileri en iyi modelleyen doğrunun formülünü çıkartabilmeyi amaçlar. Buradaki hata miktarı her noktanın doğruya olan uzaklığı olarak hesaplanabilir. ‘y’ değerleri tahmin

edilen bağımlı değişkeni ifade eder. ‘x’ değerleri girdi olarak kullandığımız bağımsız değişkeni ifade etmektedir. Veriler üzerinde çizilen doğrunun girdiler ile mesafesine bakılarak hatanın (ε_i) en aza indirilmesi amaçlanmaktadır (Eşitlik 1).

$$y_i = \alpha + \beta x_i + \varepsilon_i \quad (1)$$

Basit doğrusal regresyon ile veri kümesindeki gerçek verilere (mavi noktalar) en yakın geçen basit bir doğruyu (kırmızı doğru) sembolize etmek amaçlanmaktadır (Şekil 10).



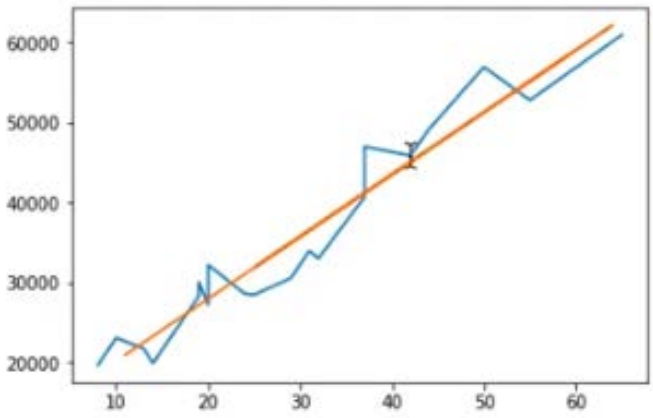
Şekil 10 Veri kümesindeki değerlere en yakın geçen doğrunun çizilmesi

Verilen veri kümesinde aylara göre satış değerleri bulunmaktadır (Şekil 11).

	Satışlar Tahmini Değerler	Satışlar Gerçek Değerler
0	20991.9	2 18865.5
1	62142.9	28 61195.5
2	32638.4	13 28540.5
3	31862	10 31609
4	58260.8	26 58484.5
5	54378.6	24 54715.5
6	58260.8	27 56317.5
7	31862	11 27897
8	38849.9	17 41544
9	50496.4	22 50651

Şekil 11 Doğrusal regresyon ile gerçek ve tahmin edilen değerlerin karşılaştırılması

Bu verilerin özelliklerine bakıldıktan sonra bir doğru formülü koyulup koyulmayacağı anlaşılabilmektedir. Minimize edilmiş bir hata miktarı ile ileriki aylardaki satışları tahmin etmek mümkün olabilmektedir (Şekil 12).



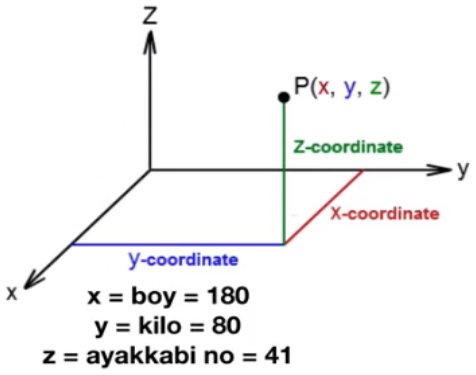
Şekil 12 Python ile basit doğrusal regresyon ile doğru çizimi

2.2.2 Çoklu Doğrusal Regresyon

Çoklu doğrusal regresyon birden fazla değişkenle çalışmak için tasarlanmıştır. Basit doğrusal regresyonda olduğu gibi çoklu doğrusal regresyonda da gerçek değerlere en yakın doğruyu en aza indirgenmiş hata ile formüle etmek amaçlanmaktadır (Eşitlik 2).

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \ldots + \beta_n x_n + \varepsilon \quad (2)$$

Veri kümesinde boy, kilo ve ayakkabı numarası gibi değerler bulunmaktadır.

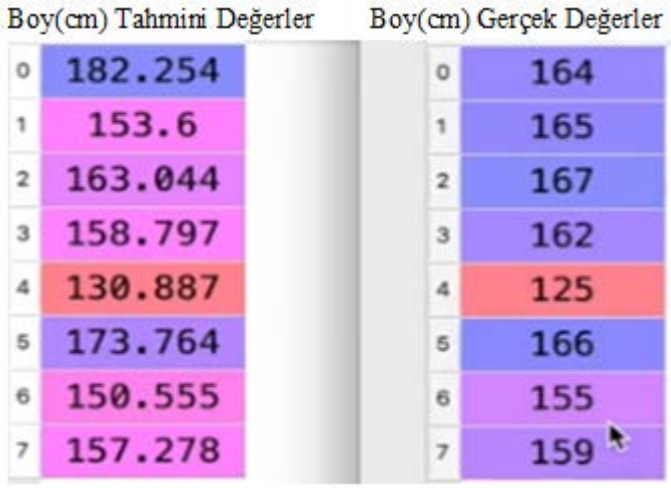


Şekil 13 Değişkenlerin 3 boyutlu uzayda gösterimi

Bir kişinin boyunu tahmini kilo, ayakkabı numarası gibi birden fazla değişkene bağlı olduğunda çoklu doğrusal regresyon ile bir doğru formülü elde edilmektedir (Eşitlik 3).

$$Boy = a + b(kilo) + c(ayakkabı\ no) + \varepsilon \quad (3)$$

Sadece kilo ve ayakkabı numarası değişkenleri kullanarak python üzerinde bu formülü uyguladığımızda boy tahmini için belli bir hata miktarı ile gerçek değerlere yakın sonuçlar elde edildiği görülmektedir (Şekil 14).



	Boy(cm) Tahmini Değerler		Boy(cm) Gerçek Değerler
0	182.254	0	164
1	153.6	1	165
2	163.044	2	167
3	158.797	3	162
4	130.887	4	125
5	173.764	5	166
6	150.555	6	155
7	157.278	7	159

Şekil 14 Python da çoklu regresyon ile gerçek ve tahmin edilen değerlerin karşılaştırılması

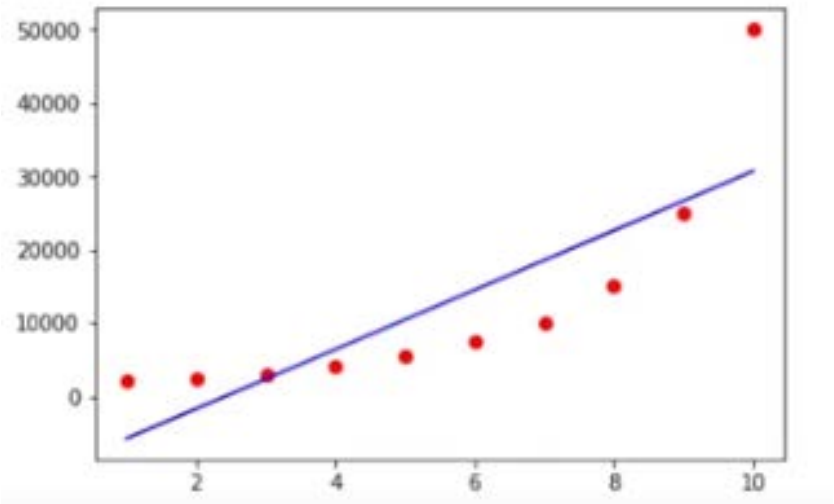
Örnek veri kümesinde kilo ve ayakkabı numarası değerleri kullanılmıştır. Ancak boy değerine etki eden başka değişkenler eklenerek daha doğru tahminler elde edilebilir.

2.2.3 Polinom Regresyon

Polinom regresyon birden fazla değişken ve bunların farklı derecelerinden oluşan bir denklemden bahsedilmektedir (Eşitlik 4).

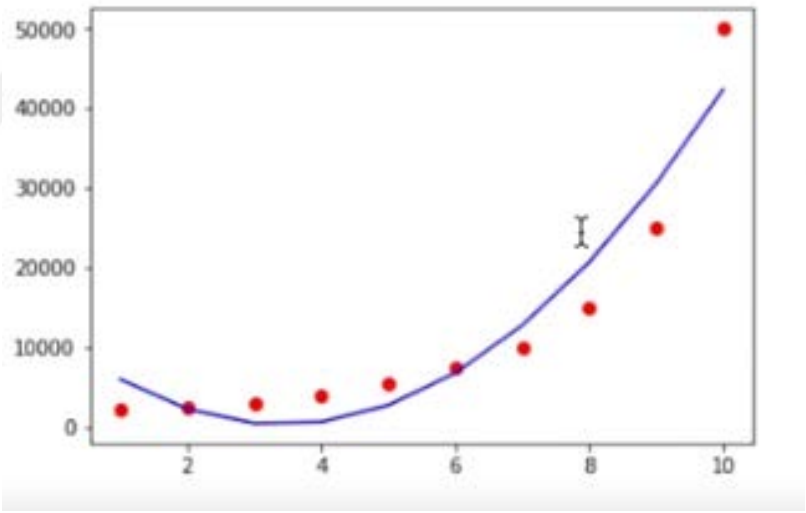
$$y = \beta_0 + \beta_1 x + \beta_2 x_1 + \beta_2 x_1^2 + \beta_3 x_2^3 + \ldots + \beta_n x^n + \varepsilon \quad (4)$$

Şekil 15 ‘te polinom dağılıma sahip bir veri kümesi için doğrusal regresyon ile bir doğru çizmeye çalışırsak çok büyük hatalar elde edileceği görülmektedir.



Şekil 15 Polinom dağılım gösteren veri kümesinin doğrusal regresyon ile çizimi

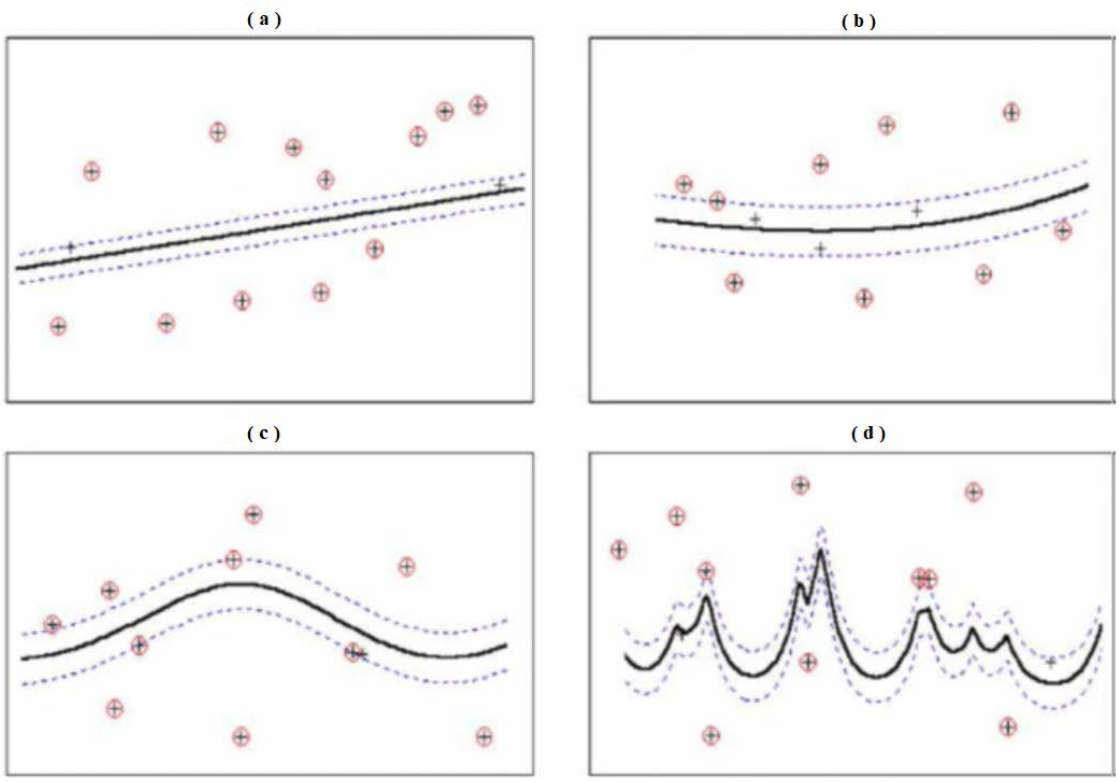
Şekil 16 ‘da polinomal regresyon ile daha önce oluşturulmuş doğrusal regresyona göre gerçek değerlere daha yakın bir polinom çizim elde edildiği görülmektedir.



Şekil 16 Polinomal dağılım gösteren veri kümesinin polinomal regresyon ile çizimi

2.2.4 Destek Vektör Regresyonu

İlk olarak destek vektörleri sınıflandırma problemleri için ortaya çıkmıştır. Doğrusal olarak birbirinden ayrılabilecek iki sınıfı birbirinden ayıran maksimum noktayı marj aralığı içine alabilen doğruları (polinom veya eğride olabilir) çizebilmeyi amaçlar (Şekil 17)



Şekil 17 Doğrusal (Linear) (a), Polinom (Polynomial) (b), Guassian (RBF) (c) ve Üssel (Exponential) (d) denklem grafikleri

Bu denklemler veri kümelerine farklı Şekilde yaklaşabilecek modeller çıkartabilmeyi sağlamaktadır. Bu Şekilde tahmin için hangi modelin kullanılacağı seçilebilmektedir.

2.2.5 Karar Ağacı (Decision Tree)

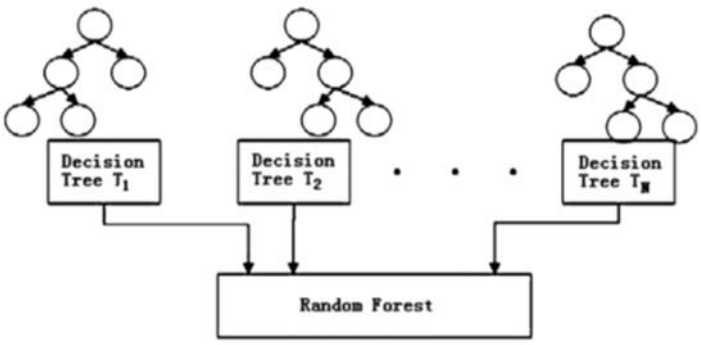
Karar ağacı algoritması ilk olarak sınıflandırma problemi için ortaya çıkmıştır. Ancak tahmin amaçlıda kullanılmaktadır. Örnek olarak boy, kilo ve yaş verilerinin olduğu bir veri kümesi ele alalım. Boy ve kilo niteliklerini kullanarak yaşı tahmin etmeyi öğretmek istediğimizde Şekil 18 ‘de ki gibi bir yapı oluşacaktır.

Şekil 18 Karar Ağacı oluşturma

Karar ağaçları oluşturulurken veri kümesinin bölme noktaları entropi ile belirlenmektedir. Basit olarak verilerin dağılımına bakılarak bölme noktaları belirlenir. Öğrenme süreci bittikten sonra Şekil 18'deki yapıya boy =180, kilo=80 verilerini girdi olarak verdiğimizde ağaç yapısını takip ederek yaşı = 31,5 olarak tahmin edeceği görülmektedir.

2.2.6 Rassal Ağaçlar (Random Forest)

Rassal ağaçlar kollektif öğrenme yöntemidir. Kollektif öğrenme birden fazla öğrenme algoritmasını aynı anda kullanarak daha başarılı sonuç elde etmeyi amaçlar. Rassal ağaçlar, veri kümesini alt parçalara bölerek her parça için farklı karar ağaçları kullanır (Şekil 19).



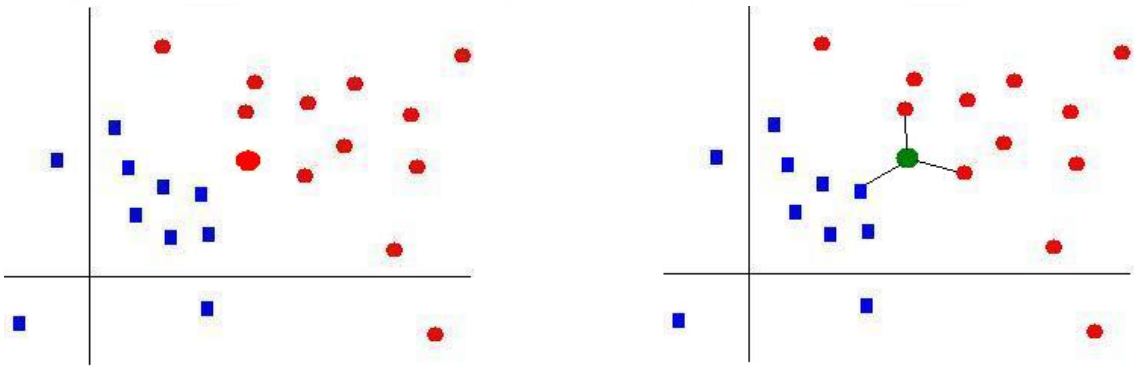
Şekil 19 Rassal Ağaçlar yapısı

Birden fazla tahmine bakılarak (majority voting(çoğunluğun oyu)) karar verilir. Karar ağaçları bildiğimiz veriler dışında yeni veriler bize veremiyorken, rassal ağaçlar her

bir veri için yeni bir sonuç döndürme özelliğine sahip olduğundan karar ağaçlarından daha başarılıdır.

2.2.7 K-NN Algoritması (K Nearest Neighborhood)

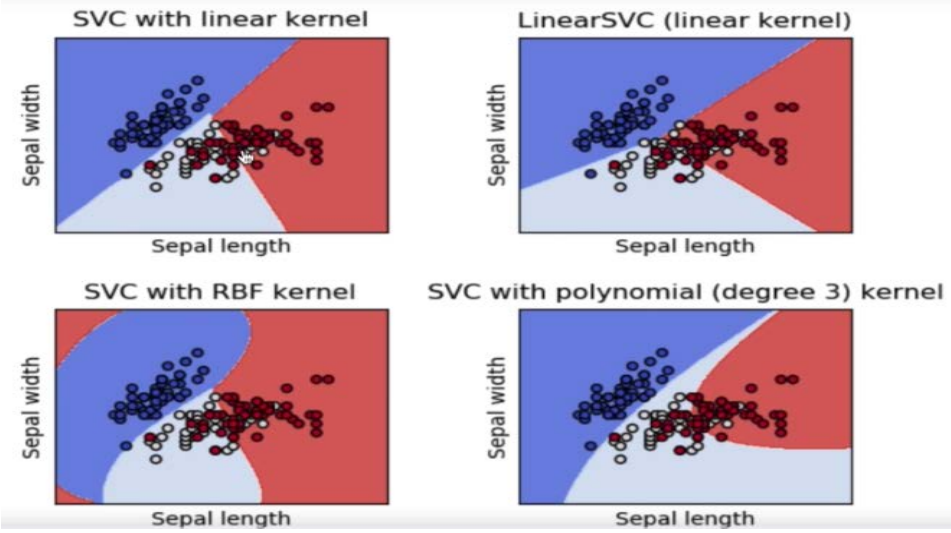
K-NN algoritması sınıflandırma problemlerinde kullanılmaktadır. Bu algoritmanın amacı yeni üyenin daha önce sınıfı belirlenmiş üyelerden belirlenen k tanesine bakılarak sınıflama işleminin yapılmasıdır. Şekil 20 'de k = 2 için sınıflandırılmak istenen yeşil üye, önceden sınıfları belli olan 2 üyeye olan mesafesine göre hangi sınıfa ait olduğuna karar verilmektedir. Önceden sınıfı belirlenen üyelere olan mesafenin bulunması için genelde Öklid Mesafesi tercih edilmektedir.



Şekil 20 Yeni gelen elemanın (yeşil nokta) hangi sınıfa dahil edileceğinin hesaplanması Şadi E. Şeker (2008)

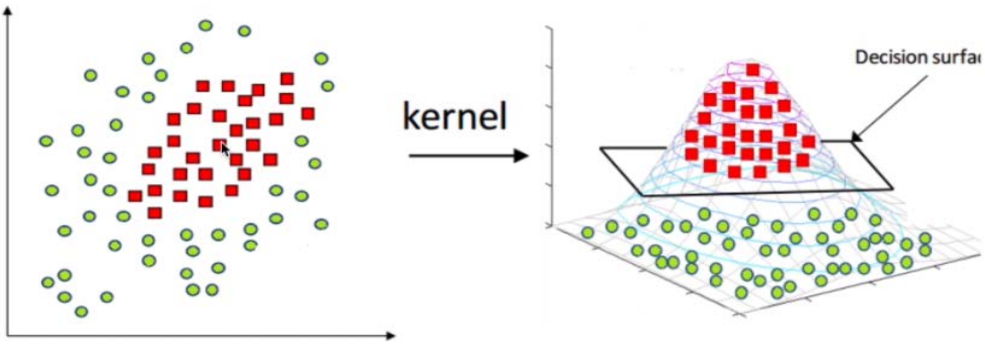
2.2.8 Destek Vektör Sınıflandırma (Support Vector Machine Classification)

Sınıflandırma algoritmaları birbirinden ayrık olan verileri sınıflandırmada gayet başarılı olmaktadır. Ancak sınır noktalara yakın olan verileri sınıflandırırken problemler çıkmaktadır. Destek vektör makinaları (SVM) en geniş margin aralığını veren durumu bulmayı amaçlamaktadır.



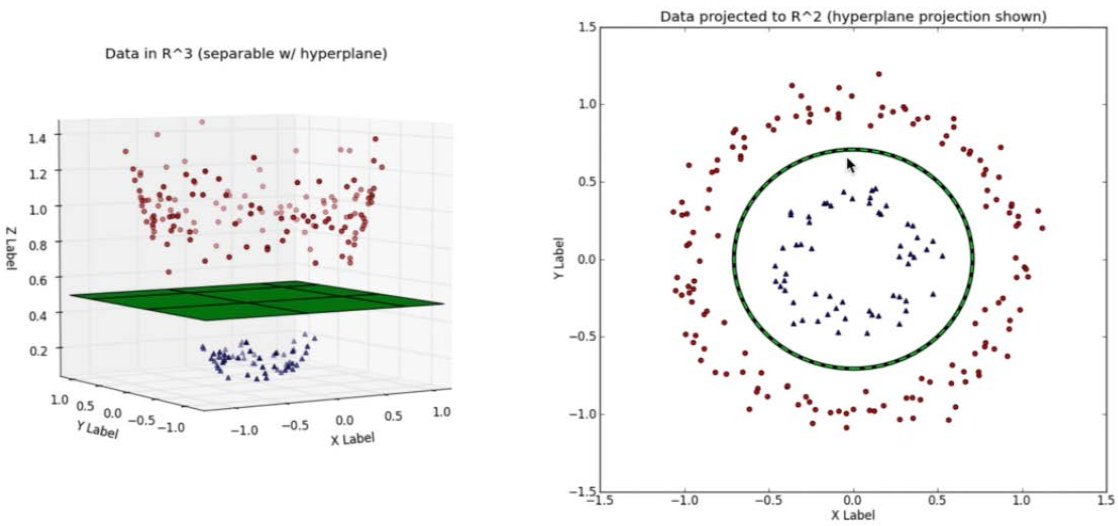
Şekil 21 İris veri kümesinin SVC ile sınıflandırılma örnekleri

Çekirdek Hilesi (Kernel Trick), doğrusal olmayan veri kümelerine boyut yükseltme (2 boyutlu bir veriyi yeni bir boyut ekleme) yapılarak çözüm bulmayı amaçlamaktadır (Şekil 22).



Şekil 22 Çekirdek Hilesi ile yeni boyut ekleme

Doğrusal olmayan bir veri kümesine SVM algoritmasını adapte edebilmek için Çekirdek hilesi kullanılabilir. Şekil 22'deki örnekte de görüldüğü gibi kırmızı noktaların orta noktasını yakalayarak boyut yükseltme yapılmıştır. Bu Şekilde Şekil 23 'teki gibi doğrusal olmayan bir sınır elde edilmektedir.



Şekil 23 SVM ve Çekirdek Hilesi ile doğrusal olmayan sınır elde edilmesi

2.2.9 Naif Bayes (Naive Bayes)

Verilen bilgilerden özellik vektörleri oluşturarak eğitim gerçekleştirerek bu eğitim sonucunda oluşan model ile doğru sınıflandırmayı yapmayı amaçlamaktadır. Sınıflandırma algoritması olan Naif Bayes olasılıksal dağılımlara bakarak sonuç üretmektedir. Naif Bayes sınıflandırıcısı, sınıflandırma görevi için kullanılan olasılıklı bir makine öğrenme modelidir. Sınıflandırıcının noktası, Bayes teoremine dayanmaktadır (Eşitlik 5).

$$P(C_i \mid X) = \frac{P(C_i \mid X)\, P(C_i)}{P(X)} \qquad (5)$$

Multinomial, Bernoulli ve Gauss Naif Bayes çeşitleri bulunan Naif Bayes algoritmaları çoğunlukla duyarlılık analizinde, spam filtrelemede, öneri sistemlerinde kullanılmaktadır. Uygulamaları hızlı ve kolaydır ancak en büyük dezavantajı, tahmincilerin bağımsız olma gerekliliğidir. Gerçek yaşam durumlarında, öngörücüler bağımlıdır. Bu da sınıflandırıcının performansını etkilemektedir.

2.3 Derin Öğrenme ve Doğal Dil İşleme

Derin öğrenme el ile çok az mühendislikle büyük miktarda hesaplama ve veri toplamanın bir yolunu sunar (LeCun, Bengio, & Hinton, 2015). Son DDİ araştırmaları giderek yeni derin öğrenme yöntemlerinin kullanımına odaklanmaktadır. Onlarca yıldır

DDİ sorunlarını hedef alan makine öğrenme yaklaşımları, çok yüksek boyutlu ve seyrek özellikler üzerine eğitilmiş sığ modellere dayanmaktadır. Son birkaç yılda, yoğun vektör temsillerine dayanan sinir ağları, çeşitli DDİ görevlerinde üstün sonuçlar vermektedir. Bu eğilim, kelime gömme ve derin öğrenme yöntemlerinin başarısı ile ortaya çıkmıştır (Mikolov, Karafiat, Burget, Cernocky, & Khudanpur, 2010) (Mikolov, Sutskever, Chen, Corrado, & Dean, 2013). Derin öğrenme, çok seviyeli otomatik özellik sunum öğrenmesini sağlar. Buna karşılık, geleneksel makine öğrenmeye dayalı DDİ sistemleri, el yapımı özelliklere dayanmaktadır. Bu el yapımı özellikler zaman alıcı ve çoğu zaman eksik kalmaktadır (Young, Hazarika, Poria, & Cambria, 2018).

DDİ, insan dilinin otomatik analizi ve gösterimi için teorik olarak motive edilmiş bir hesaplama teknikleri yelpazesidir. DDİ araştırması, makine çevirisi, bilgi alma, metin özetleme, soru cevaplama, bilgi çıkarma, konu modelleme ve daha yakın zamanda fikir madenciliği gibi görevlere odaklanmaktadır. İlk günlerde yapılan çoğu DDİ araştırması, kısmen sözdizimsel işlemlerin açıkça gerekli olduğu ve kısmen sözdizimine dayalı işlem fikrinin örtük veya açık bir Şekilde onaylanması nedeniyle, sözdizimine odaklanmıştır. DDİ araştırması başlamasından bu yana, bir cümle analizinin 7 dakika sürebildiği işlemlerden, minyonlarca web sayfasının bir saniyeden daha kısa sürede işlenebildiği çağa evirilmiştir (Cambria & White, 2014).

DDİ, bilgisayarların ayrıştırma ve konuşma bölümünün (POS) etiketlemesinden makine çevirisi ve diyalog sistemlerine kadar her seviyede doğal dille ilgili çok çeşitli görevleri gerçekleştirmesini sağlar. Derin öğrenme mimarileri ve algoritmaları, bilgisayar görüşü ve örüntü tanıma gibi alanlarda zaten etkileyici gelişmeler sağlamıştır (Young, Hazarika, Poria, & Cambria, 2018).

Collobert ve arkadaşları (2011), basit ve derin bir öğrenme çerçevesinin, Yapay sinir ağlarındaki (YSA) adlandırılmış varlık tanıma (NER), semantik rol etiketleme (SRL) ve POS etiketleme gibi birçok DDİ görevinde en modern yaklaşımlardan daha iyi performans gösterdiğini göstermiştir. O zamandan beri zor DDİ görevlerini çözmek için Konvolüsyonel Sinir Ağları (CNN'ler) ve Tekrarlayan Sinir Ağları (RNN'ler) gibi doğal dil görevlerine uygulanan temel derin öğrenme ile ilgili modeller ve yöntemler önerilmiştir (Young, Hazarika, Poria, & Cambria, 2018).

Peilu Wang ve arkadaşlarının yaptığı çalışmada, konuşma parçası (POS) etiketleme görevi için sözcük gömme ile BLSTM-RNN kullanılması önerilmiştir. Penn Treebank WSJ test setinde test edildiğinde, 97.40 etiketleme hassasiyetinde son teknoloji performans elde edilmiştir. İki Yönlü Uzun Kısa Süreli Bellek Tekrarlayan Sinir Ağının (BLSTM-RNN) sıralı verileri, konuşma ifadelerini veya el yazısı gibi belgeleri etiketlemek için çok etkili olduğu gösterilmiştir (Wang, Qian, K. Soong, He, & Zhao, 2015).

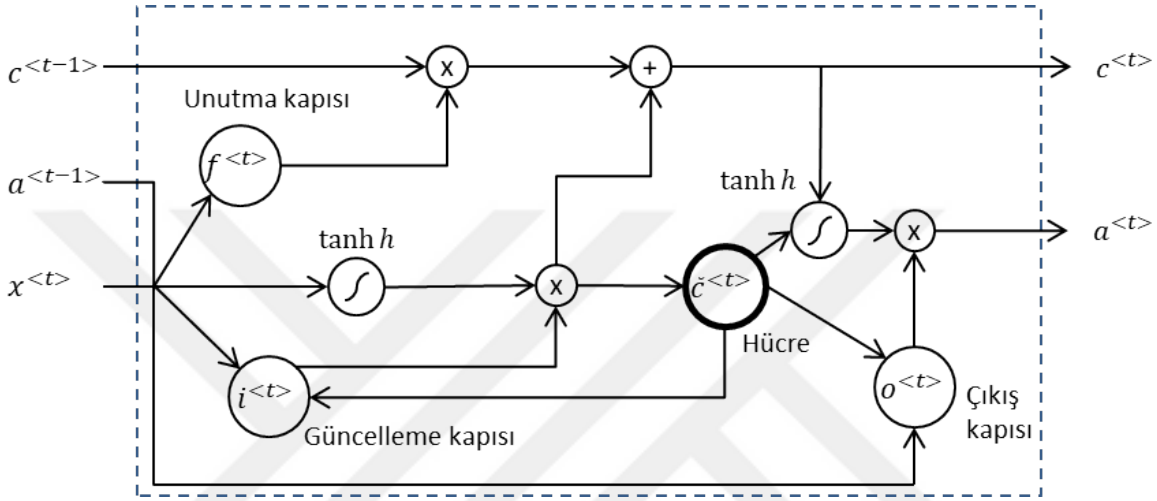
2.4 Tekrarlayan Sinir Ağları (RNN)

RNN, sıralı verileri işlemek için bir sinir ağları ailesidir (Elman, 1990). "tekrarlayan" terimi, çıktının önceki hesaplamalara ve sonuçlara bağlı olarak dizinin her örneği üzerinde aynı görevi gerçekleştirdiği için kullanılmaktadır. Genellikle, belirteçleri birer birer besleyerek bir sekansı temsil etmek için sabit boyutlu bir vektör üretilmektedir. RNN'ler önceki hesaplamalar üzerinde bir hafıza bilgisine sahiptir ve bu bilgi mevcut işlemler üzerinde kullanılmaktadır. Bu model, dil modellemesi, makine çevirisi, konuşma tanıma, resim yazısı gibi birçok DDİ görevi için uygun görülmektedir. Bu, RNN'lerin son yıllarda DDİ uygulamaları için giderek daha popüler olmasını sağlamıştır (Young, Hazarika, Poria, & Cambria, 2018).

RNN'nin dizi modelleme görevlerine uygunluğuna yardımcı olan diğer bir faktör, çok uzun cümleler, paragraflar ve hatta belgeler dahil olmak üzere değişken metinleri modelleme kabiliyetinden kaynaklanmaktadır (Tang, Qin, & Liu, 2015). Diğer sinir ağlarının aksine RNN'ler daha iyi modelleme yeteneği sağlayan ve sınırlandırılmamış bağlamı yakalama olanağını yaratan esnek hesaplama adımlarına izin vermektedir. Keyfi uzunluk girişine izin verme yeteneği, RNN'lerin popülerliğine etki eden en önemli özellik olmuştur (Chung, Gulcehre, Cho, & Bengio, 2014). Bununla birlikte, pratikte, bu basit RNN ağlarında, ağdaki önceki katmanların parametrelerinin öğrenilmesini ve ayarlanmasını gerçekten zorlaştıran, kaybolan gradyan problemi ortaya çıkmaktadır. Bu problem Sinir ağlarına daha fazla katman eklendiğinden, kayıp fonksiyonunun gradyanlarının sıfıra yaklaşmasına sebep olur ve bu da ağı eğitmeyi zorlaştırmaktadır. DDİ uygulamalarında LSTM gibi çeşitli ağlar ile bu sınırlamanın üstesinden gelinmiştir (Young, Hazarika, Poria, & Cambria, 2018).

2.4.1 Uzun-Kısa Vadeli Bellek (LSTM)

LSTM'ler uzun vadeli bağımlılık problemini çözmek için tasarlanan özel bir RNN türüdür (Hochreiter & Schmidhuber, 1997). Bilgiyi uzun süre hatırlama davranışı sergilerler. LSTM'ler tüm RNN'ler gibi bir sinir ağının tekrar eden hücrelerine sahiptir, fakat tekrar eden hücre tek bir sinir ağı kapısına sahip olmak yerine, etkileşime giren 3 adet kapıya (Şekil 1) sahiptir (Graves & Schmidhuberab, 2005).



Şekil 24 Uzun-Kısa Vadeli Hafıza Hücresi

Bir LSTM davranışını tanımlayan kapılar güncelleme (update), unutma (forget) ve çıkış (output) kapıları olarak adlandırılmaktadır. Burada $i^{<t>}$, $f^{<t>}$ ve $o^{<t>}$ sırasıyla t anındaki güncelleme, unutma ve çıkış kapılarını, $c^{<t>}$, hücre durumunu ve $a^{<t>}$, tüm işlemler sonunda karar verilen bilgiyi belirtir.

Bir sigmoid(σ) işlevine sahip unutma kapısı (Eşitlik 6) giriş ($x^{<t>}$) ve önceki hücre durumu ($a^{<t-1>}$) bilgisine bakarak bilginin unutulup unutulmayacağına karar verir.

$$f^{<t>} = \sigma(W_f[a^{<t-1>}, x^{<t>}]) + b_f \quad (6)$$

Bir sigmoid(σ) işlevine sahip güncelleme kapısı (Eşitlik 7) hangi bilgilerin güncelleneceğine karar verir.

$$i^{<t>} = \sigma(W_u[a^{<t-1>}, x^{<t>}]) + b_u \quad (7)$$

Bir $\tan h$ işlevi, hücre durumuna ($c^{<t>}$) eklenebilecek yeni hücre durumu aday bilgileri ($\tilde{c}^{<t>}$) (Eşitlik 8) vektörünü oluşturur.

$$\tilde{c}^{<t>} = \tan h(W_c[a^{<t-1>}, x^{<t>}]) + b_c \quad (8)$$

Dolayısıyla, mevcut hücre durumu ($c^{<t>}$) (Eşitlik 9), hem önceki hücre durumu ($\tilde{c}^{<t>}$) hem de hücre tarafından üretilen mevcut bilgileri kullanılarak elde edilmiş olur.

$$c^{<t>} = i^{<t>} * \tilde{c}^{<t>} + f^{<t>} * c^{<t>} \quad (9)$$

Bir sigmoid(σ) işlevine sahip çıkış kapısı (Eşitlik 10) hücre durumunun hangi kısımlarını üreteceğimize karar verir. Daha sonra hücre durumu ($c^{<t>}$) $\tan h$ işlevi ile filtrelenir ve çıkış kapısının çıktısı ile çarpılır. Bu durumda sadece karar verilen bilgiler ($a^{<t>}$) (Eşitlik 11) elde edilir.

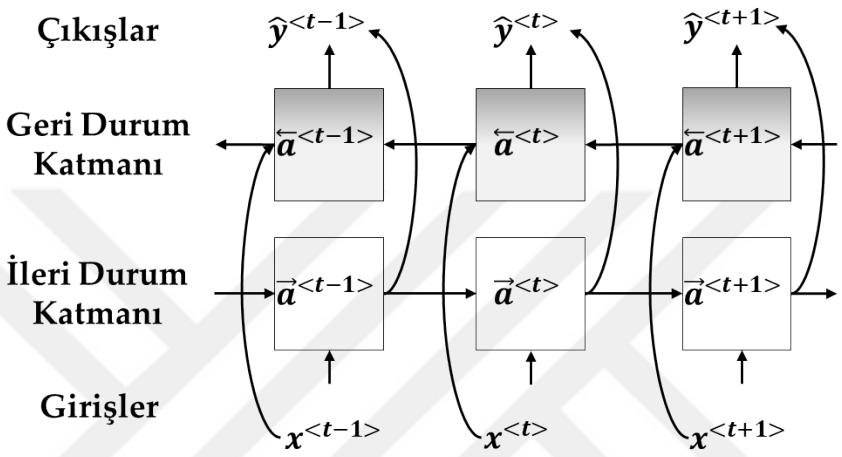
$$o^{<t>} = \sigma(W_o[a^{<t-1>}, x^{<t>}]) + b_o \quad (10)$$

$$a^{<t>} = o^{<t>} \tan h(c^{<t>}) \quad (11)$$

Standart LSTM ağları dizileri geçici sırayla işler, gelecekteki bağlamı görmezden gelirler. (Schuster & K. Paliwal, 1997)'da önerilen iki yönlü RNN gizli bağlantıların ileriye doğru sırayla aktığı, ikinci bir katman uygulayarak tek yönlü LSTM ağlarını genişletmektedir.

2.4.2 İki Yönlü Uzun-Kısa Vadeli Bellek (BLSTM)

İki Yönlü Tekrarlayan Sinir Ağları (BRNN) bir dizinin hem önceki zamanlardaki hem de ileriki zamanlardaki bilgilerini kullanabilmeyi sağlar. Bu yöntem, normal bir RNN'nin durum nöronlarını pozitif zaman yönünden (ileri durumlar- $\vec{a}$) ve negatif zaman yönünden (geri durumlar- $\overleftarrow{a}$) birbiriyle bağlantısı olmayan iki duruma (Şekil 2) bölmektedir (Schuster & K. Paliwal, 1997).

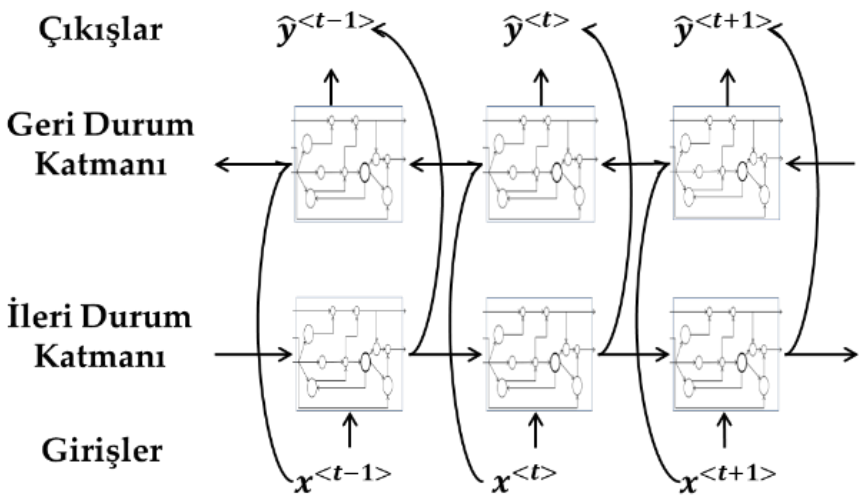


Şekil 25 İki Yönlü Tekrarlayan Sinir Ağının (BRNN) 3 Zaman Adımı

Böylece normal bir RNN'den farklı olarak iki katmandan gelen değerler ile hesaplama yapılır. Geri durumlar ve ileri durumlardan elde edilen değerler, ağırlıklar ve koşullu olasılık bayes değerlerinin hesaplanmasıyla t anındaki $\hat{y}^{<t>}$ değeri (Eşitlik 12) elde edilmiş olur.

$$\hat{y}^{<t>} = g(W_y[\vec{a}^{<t>}, \overleftarrow{a}^{<t>}]) + b_y \qquad (12)$$

BRNN 'lerde tekrar eden hücrelerde LSTM hücresi kullanıldığında İki yönlü uzun-kısa vadeli bellek ağı (BLSTM) mimarisi (Şekil 3) elde edilmektedir (Graves & Schmidhuberab, 2005).



Şekil 26 İki Yönlü Uzun-Kısa Vadeli Bellek Ağının ileri ve geri yönde zaman dizisi

Bu mimaride ileri ve geri yöndeki hesaplamalar aynı anda çalıştırılır ve iki yönde yapılan hesaplamalar sonucu ulaşılan bilgiler birleştirilerek çıktı sonucuna ulaşılır. Bu Şekilde iki yöndeki bilgilerin kullanılması sıralı verilerin işlenmesinde avantaj sağlar. (Graves & Schmidhuberab, 2005)'deki araştırmada iki yönlü ağların tek yönlü ağlardan daha etkili olduğunu gösterilmiştir. İki yönlü uzun-kısa vadeli bellek (BLSTM) ağları, doğal dil işleme çalışmalarında da başarılı Şekilde kullanılmaktadır (Wang, Qian, K. Soong, He, & Zhao, 2015).

Günümüzde DDİ alanında çalışan araştırmacılar, derin öğrenme yöntemlerinin kullanımına yönelmektedir. Derin öğrenme yöntemleri kullanılarak geliştirilen doğal dil işleme çalışmalarının İngilizce üzerine yoğunlaştığı görülmektedir (Şeker, Dirib, & Balık, 2017). Türkçe için derin öğrenme yöntemleri ile doğal dil işleme çalışmaları sınırlıdır. Türkçe sözcük etiketleme için derin öğrenme ile eğitilmiş bir model henüz bulunmamaktadır. Bu çalışmada, RNN mimarisinde iki yönlü uzun-kısa vadeli bellek kullanılarak Türkçe sözcük etiketleme için bir DDİ modelinin geliştirilmesi amaçlanmıştır. Platform ile dil bilim bilimi ve bilgisayar bilimi altında çalışan araştırmacılar için kaynak eksikliğinin giderilmesi hedeflenmektedir. Ayrıca platformun, insana özgü doğal dile yakın seviyede işleme gerçekleştirmeyi destekleyeceği düşünülmektedir.

3. GEREÇ ve YÖNTEM

Bu çalışmada Keras, TensorFlow ve BLSTM-RNN derin öğrenme modeli kullanılmıştır. Yazılım geliştirilmesi için Python programlama dili kullanılmıştır. Tez çalışmasında tavsiye edilen BLSTM yapay sinir ağı modeli ve etiketli veri seti üzerinde değişik kombinasyonlar denenerek sistemin başarımına etkisi gözlemlenmiştir.

BLSTM-RNN ile eğitimi gerçekleştirilen derlemin içeriğinde (Ozturk, Sankur, Gungor, & Yilmaz, 2014)'deki çalışma ile hazırlanan belgelerin sadece özet kısımları kullanılmıştır.

3.1 Veri Seti

BLSTM-RNN ile eğitimi gerçekleştirilen derlemin içeriğinde 34 konu ile işaretlenmiş, konu başına yaklaşık 200 belgeden oluşan, toplam 7312 adet belge bulunmaktadır. Derlem istatistikleri Tablo 1 'de gösterilmiştir.

Tablo 1 Derlem istatistik tablosu

İstatistik	Miktar
Toplam Sözcük	1.186.329
Toplam Konu	34
Toplam Belge	7.312

Farklı alanlarda Türkçe dergi ve konferansların bildiri kitaplarından derlenen. (Ozturk, Sankur, Gungor, & Yilmaz, 2014)'deki çalışma ile toplanmış belgelerin özet kısımlarından oluşan veri seti oluşturulmuştur. Tablo 2'de detayları verilen veri seti, Antropoloji, Arkeoloji, Coğrafya, Dil Bilim, Dini Araştırmalar, Eğitim Bilimleri, Ekonomi, Ekonometri, Felsefe, İletişim, Kütüphanecilik, Siyasal Bilimler, Sosyoloji, Tarih, Turizm, Borsa-Bankacılık, Harici Tıp, Dahili Tıp, Temel Tıp, Eczacılık, Biyoloji, Çevre Bilimleri, Hayvancılık, Spor Bilimleri, İşaret İşleme, Elektronik İletişim, Endüstri Mühendisliği, Makine Mühendisliği, İnşaat Mühendisliği, Gıda Mühendisliği, Biyomedikal Mühendisliği, Jeoloji Mühendisliği, Mimarlık ve Hukuk olmak üzere ULAKBİM Ulusal Veri Tabanları'nın 5 ana kategorisinden konuları kapsamaktadır.

Veri setinde bulunan konulara göre belge ve sözcük sayısı Tablo 2 'de gösterilmektedir.

Tablo 2 Konulara göre belge ve sözcük sayısı tablosu

Konu	Belge Sayısı	Sözcük Sayısı
Antropoloji	200	23.520
Arkeoloji	239	128.222
Coğrafya	200	31.466
Dil Bilim	175	24.378
Dini Araştırmalar	208	23.768
Eğitim Bilimleri	196	31.242
Ekonomi	210	28.970
Ekonometri	200	24.442
Felsefe	209	26.426
İletişim	211	39.092
Kütüphanecilik	182	25.521
Siyasal Bilimler	216	24.563
Sosyoloji	250	33.435
Tarih	205	31.984
Turizm	216	36.877
Borsa-Bankacılık	149	19.064
Harici Tıp	200	39.962
Dahili Tıp	200	33.959
Temel Tıp	200	42.350
Eczacılık	200	26.392
Biyoloji	173	23.383
Çevre Bilimleri	199	37.576
Hayvancılık	205	37.041
Spor Bilimleri	200	40.269
İşaret İşleme	638	71.956
Elektronik İletişim	185	20.320
Endüstri Mühendisliği	191	29.820
Makine Mühendisliği	200	28.719
İnşaat Mühendisliği	200	27.935
Gıda Mühendisliği	200	33.143
Biyomedikal Mühendisliği	210	29.744
Jeoloji Mühendisliği	199	46.756
Mimarlık	200	33.995
Hukuk	246	30.039

3.2 Etiket Seçimi

Birçok dil için diller arası tutarlılığı sağlamak için Evrensel Bağımlılıklar (Universal Dependencies) Türkçe sözcük etiketleri 1. sürüm kullanılmıştır (Universal Dependencies, 2019). Yapay sinir ağının eğitiminin gerçekleştirilmesi için ihtiyaç duyulan etiketli veri bir trigram HMM POS Etiketleyici kullanılarak ilk etiketlemesi gerçekleştirilmiştir.

Etiketlemesi gerçekleştirilen veri setinde bulunan etiket türleri ve dağılımları tablo 3'te gösterilmiştir.

Tablo 3 Sözcük türlerine göre analiz sonuçları

Sözcük Türü	Sözcük Etiketi	Etiket Sayısı
İSİM	NOUN	343.637
FİİL	VERB	46.583
SIFAT	ADJ	86.996
ZARF	ADV	23.757
ZAMİR	PRON	4.368
BAĞLAÇ	CONJ	37.616
SAYISAL	NUM	25.255
NOKTALAMA	PUNC	112.043
ÜNLEM	INTJ	916
BİLİNMEYEN	X	101.875

3.3 Deney Ortam Kurulumu

Deneyler NVIDIA Geforce GTX 950M (2GB) GPU, Intel Core, 2.60GHz sekiz çekirdekli CPU, 12 Gb bellekli, Windows 10 işletim sistemi kullanan bilgisayar üzerinde gerçekleştirilmiştir. Deneylerin Yapılacağı ortam kurulumu için aşağıdaki adımlar izlenmiştir.

a) Windows'a Anaconda kurulumu gerçekleştirilmiştir.
b) Python için açık kaynak bir IDE olan Spyder kurulumu gerçekleştirilmiştir.
c) Python 3.6 kurulumu gerçekleştirilmiştir.
d) Anaconda komut isteminden Theano kurulumu gerçekleştirilmiştir.
e) Anaconda komut isteminden Tensorflow kurulumu gerçekleştirilmiştir.
f) Anaconda komut isteminden Keras kurulumu gerçekleştirilmiştir.

Keras kütüphanesinin kelimelerle veya etiketlerle değil sayılarla çalışması gerekmektedir. Her bir kelimeye ve etikete benzersiz bir tamsayı atanarak bir sözlükte dizine alınmıştır. Bu sözlükler kelime dağarcığı ve etiket dağarcığı olarak adlandırılabilir. Ayrıca bilinmeyen kelimeler için bir değer (OOV -Out Of Vocabulary) eklenmiştir. Keras yalnızca sabit boyutlu dizilerle ilgilendiği için veri setindeki en uzun cümle hesaplanmıştır. Buna göre diğer cümlelerde aynı boyutu sağlamak için bir değer (indeks olarak "0" ve karşılık gelen etiket olarak "-PAD-") eklenmiştir. Son olarak eğitim aşamasına geçilmeden önce Etiket dizileri One-Hot Encoded etiketlerinin dizilerine dönüştürülmüştür.

3.4 Eğitim ve Test verisi

Oluşturulan yapay sinir ağının öğrenmesinin gerçeklenmesi için eğitim verisine ve yapay sinir ağının ne kadar öğrendiğini anlamak için test verisine ihtiyaç duyulmaktadır. Eğitim verisi önceden etiketlenmiş kelimelerden oluşmaktadır. Test verisi, eğitim verisi içinde bulunmayan ve etiketli olmayan kelimelerden oluşmaktadır. Yapay sinir ağının eğitim verisi ile öğrenimi gerçekleştirilerek bir model oluşturulmaktadır. Daha sonra test verisi modele girdi olarak verilerek kelimelerin etiketlerinin tahmin etme başarısı ölçülmektedir.

Bu tez çalışması için toplanan özet kısımlar, standart bir trigram Saklı Markof Model konuşma parçacığı etiketleyici (HMM POS Tagger) ile eğitim verisinin sözcük etiketlemesi gerçekleştirilmiştir (Çöltekin, 2014). Elde edilen veriler NLTK Kütüphanesi kullanılarak büyük harf, sayılar ve noktalama işaretlerinden temizlenerek eğitime hazır forma dönüştürülmüştür.

İdeal sonuçlar sağlayan parçalama kuralı kabul edilen 2/3 yüzde oranı (Dobbin & Simon, 2011) kullanılarak veriler, eğitim verileri (%66) ve test verileri (%33) olarak ayrılmıştır. Veri bölme durumu rastgele olarak ayarlanmıştır. Scikit-Learn Kütüphanesi çapraz doğrulama (cross validation) yöntemi ile başarı durumu gözlenmiştir.

Çıkışları olasılık olarak yorumlayabilmek ve çapraz doğrulama ile başarı durumu gözlemleyebilmek için yapay sinir ağının son katmanında Softmax aktivasyon katmanı (Eşitlik 13) kullanılmıştır (Dunne & Campbell, 1997).

$$\sigma(Z)_j = \frac{e^{Z_j}}{\sum_{k=1}^{K} e^{Z_k}} \qquad (13)$$

Softmax katmanındaki hatayı ölçmek için Softmax çapraz entropi kaybı (categorical cross entropy) (Eşitlik 14) kullanılmıştır (Dunne & Campbell, 1997)

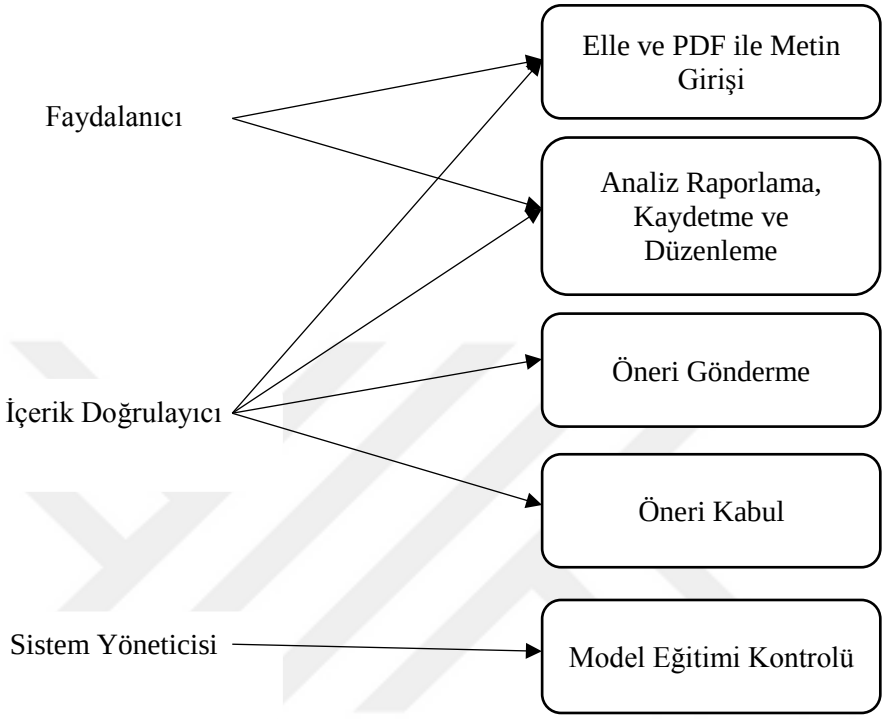
$$L(w) = -\frac{1}{N}\sum_{n=1}^{N}[y_n \log \hat{y}_n + (1 - y_n)\log(1 - \hat{y}_n)] \qquad (14)$$

Derin sinir ağlarının eğitimi için özel olarak tasarlanmış az bellek gerektiren ve hesaplama açısından verimli bir uyarlamalı öğrenme hızı optimizasyon algoritması olan Adam Optimizasyon metodu (Kingma & Ba, 2015) (Eşitlik 15) kullanılmıştır.

$$w_t = w_{t-1} - \eta \frac{\widehat{m}_t}{\sqrt{\widehat{V}_t + \epsilon}} \qquad (15)$$

4. AKILLI DERLEM

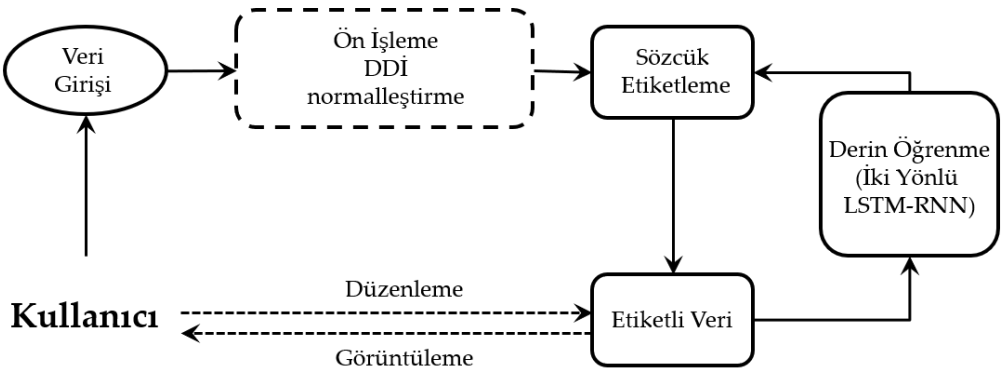
Bu çalışmada geliştirilen Akıllı Derlem platformunun Use Case diyagramı Şekil 27'de sunulmaktadır. Rollere göre yetki değişiklikleri gösteren platformda, faydalanıcı ve içerik doğrulayıcı ve sistem yöneticisi yer almaktadır.



Şekil 27 Use Case diyagramı

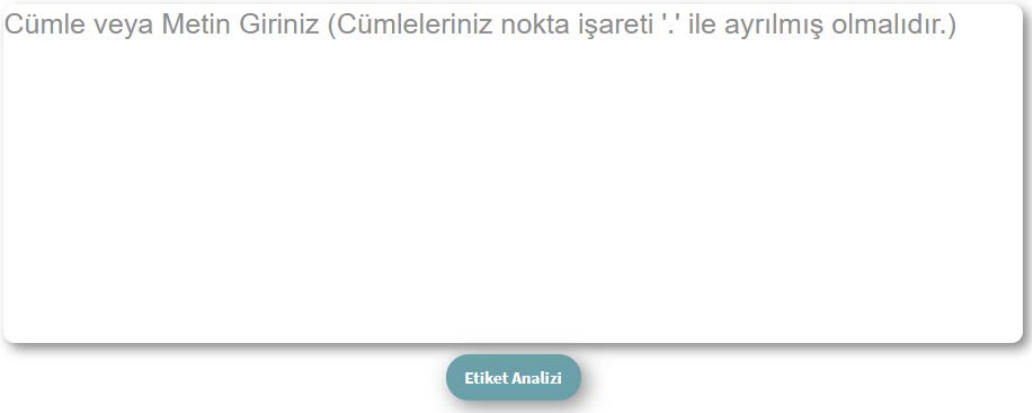
Faydalanıcı, metin girişi ile etiket analizi yapabilir, yapmış olduğu çalışmaları kaydedebilir ve daha sonra kayıtlı çalışmalarını raporlayabilir ve düzenleyebilir. İçerik doğrulayıcı faydalanıcının sahip olduğu yetkilere sahiptir. Bunların dışında yapmış olduğu analizlerde gördüğü yanlış etiketlemeleri düzelterek öneri gönderebilir. Ayrıca diğer içerik doğrulayıcılar tarafından gönderilen önerileri inceleyip doğrulayabilir. Sistem yöneticisi edilen önerilerin uygun formatta birleştirilmesini yaparak planlanan zaman diliminde eğitim verisine eklenmesini sağlayarak Akıllı Derlemin kendi kendini eğitebilmesi için doğruluğu onaylanmış etiketli girdiyi sağlar.

Önerilen sözcük etiketleme modeli, doğal dil araştırmacılarına, kendi analizlerini gerçekleştirme ve kullanabilme imkânı verecek bir platform olan Akıllı Derlem ile sunulmaktadır. Dinamik ve kullanımı kolay bir web tabanlı platform olarak geliştirilen Akıllı Derlem platformunun mimarisi ve çalışma şekli sunulmaktadır (Şekil 28).



Şekil 28 Akıllı Derlem Türkçe Doğal Dil İşleme Platformu

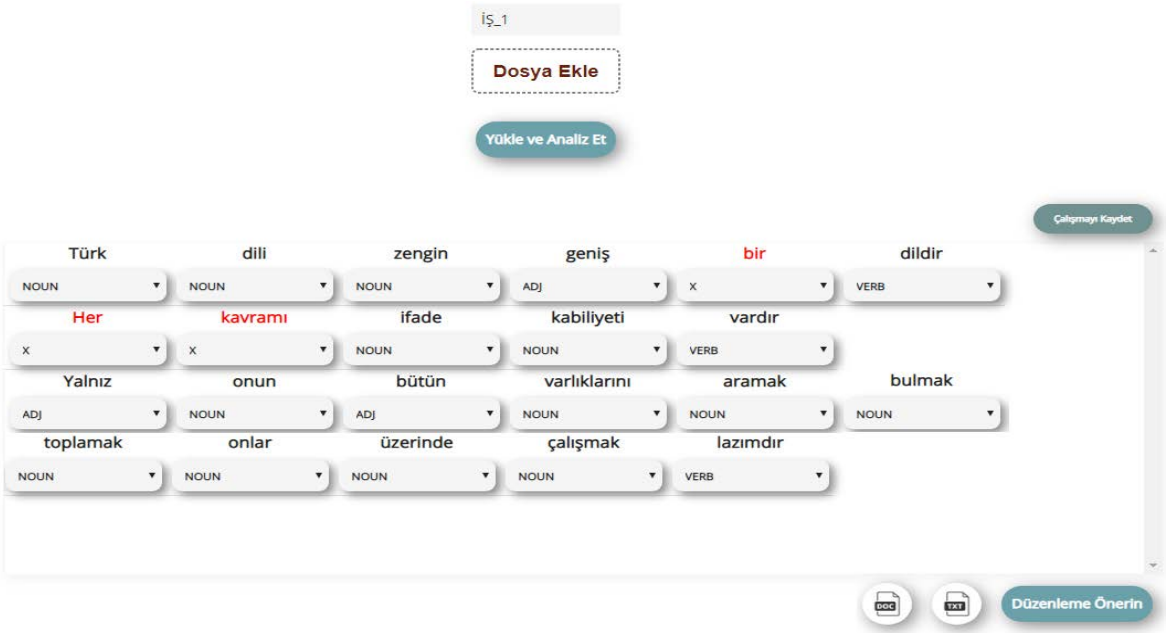
Akıllı Derlem, araştırmacılara kaynak sağlaması ve kendi kendini geliştirmesi için kullanıcılara metin girdisi ve belge girdisi (PDF) olmak üzere 2 ayrı ekrandan veri girişi sağlamaktadır. Metin analizi ekranı, kullanıcılara el ile metin girme ve analiz gerçekleştirme imkânı sağlamaktadır. Belge analizi ekranı ile kullanıcılar PDF formatında belgeler yükleyerek verilerin analizlerini gerçekleştirebilmektedir (Şekil 29,30).



Şekil 29 Metin Analizi ekranı

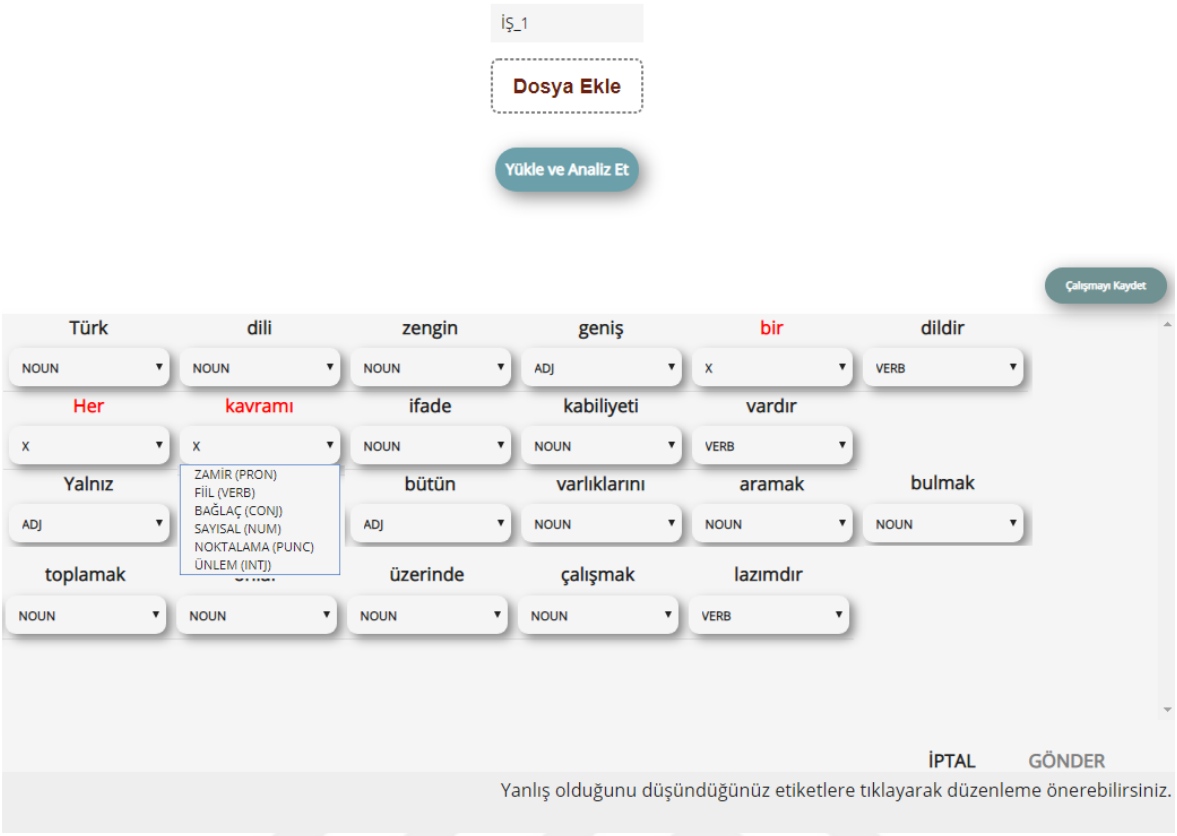
Şekil 30 PDF Analizi ekranı.

Şekil 29 ve Şekil 30 da gösterilen ekranlardan alınan veriler ilk olarak ön işleme sürecinden geçmektedir. Ön işleme aşamasında yazılı belgelerden elde edilen verilerin işlenebilir hale getirilmesi için DDİ normalleştirme süreci uygulanmaktadır. Normalleştirme süreci küçük harfe dönüştürme, noktalama işaretlerinden arındırma ve sözcük parçalama (tokenize) işlemleridir. Veri ilk olarak küçük harfe dönüştürülmektedir. İkinci aşamada, sözcük parçalama ile veriden cümleler elde edilmektedir. Son olarak da veri, cümle içerisinde noktalama işaretlerinden arındırılarak sözcük etiketleme için hazır hale getirilmektedir. Ön işlemeden geçen veri, sözcük etiketleme aşamasında, iki yönlü LSTM ile analiz edilmektedir. Analiz sonucu her sözcüğe cümle içerisindeki bağlamına göre en olası etiket atanmaktadır (Şekil 30).



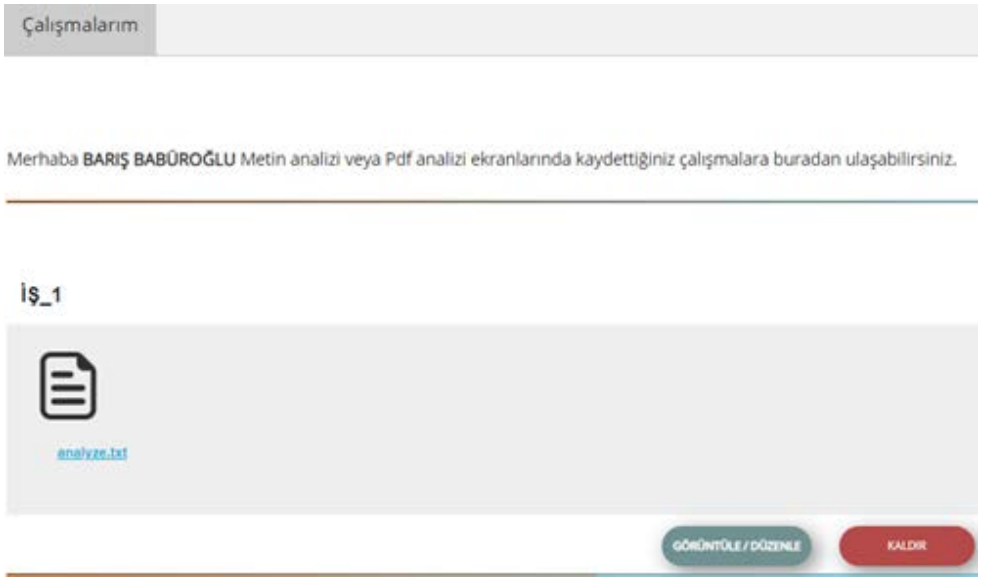
Şekil 31 Etiket analizi sonuçları

Etiketli veri kısmında, etiketi atanan sözcükler ve frekansları kullanıcıya sunulmaktadır. Gerçekleştirilen analiz sonuçlarının dökümü alınarak Türkçe doğal dil işleme çalışmalarında kullanılabilmektedir. Ayrıca kullanıcı "Düzenleme Önerin" seçeneği ile yanlış olduğunu düşündüğü etiketleri düzelterek platformun geliştirilmesine katkıda bulunabilmektedir (Şekil 31). Düzenleme yapılan etiketlenmiş veriler daha sonra eğitim aşamasına dâhil edilip etiketleme doğruluğunun arttırılması için kullanılmaktadır.



Şekil 32 Etiket Önerisi ekranı

Ayrıca Akıllı Derlem ile kullanıcının yaptığı çalışmaları kayıt altına alabilmesi sağlanmaktadır. Bu Şekilde kullanıcıya istenildiği zaman çalışmalarını düzenleme ve raporlayabilme imkânı sunulmaktadır (Şekil 32).



Şekil 33 Kayıt altına alınan çalışmalar

5. TARTIŞMA, SONUÇ ve ÖNERİLER

Bu çalışmada, derin BLSTM kullanılarak dinamik ve kullanımı kolay bir web tabanlı DDİ modeli sunulmuştur. Herhangi bir el yapımı özellik veya dış araç kullanmadan sözcüklerin bağlamsal özelliklerinden faydalanarak etiket tahminleri gerçekleştirilmiştir. Bağlama bakılarak eğitim verisinde bulunmayan sözcükler tahmin edilmiştir.

Yapay sinir ağı modelleri için aynı veri setlerini kullanılmaktadır, bu nedenle farklı sonuçlar yalnızca farklı ağlardan kaynaklanmaktadır. Modeller eğitim verileri kullanılarak eğitilmektedir ve doğrulama verilerinin performansı izlenmektedir.

Önerilen Türkçe Sözcük Etiketleme Modelinde kullanılan yapay sinir ağı modellerine ait gizli katman ve eğitim süreleri tablo 4 ‘de gösterilmiştir.

Tablo 4 BLSTM katman sayısı ve eğitim süreleri

Model	Gizli Katman	Eğitim Süresi (dk)
BLSTM	1	24
Derin BLSTM 1	2	45
Derin BLSTM 2	3	92
Derin BLSTM 3	4	118
Derin BLSTM 4	5	184
Derin BLSTM 5	6	208

Deneylerde veri setindeki cümle örneğinden 21.964 örnek eğitim için, 5.492 örnek doğrulama için kullanılarak model eğitme işlemi gerçekleştirilmiştir. Gerçekleştirilen her bir iterasyon sonunda 6.865 örnek içeren test kümesi üzerinde başarı ölçümü gerçekleştirilmiştir. Başarılı ölçüm sonuçları raporlanmıştır. Her bir model için gözlenen kayıp ve doğruluk değerleri tablo 5 ’te verilmiştir.

Tablo 5 BLSTM modellerinin başarı sonuçları

Model	Gizli Katman	Kayıp (Loss)	Doğruluk (Accuracy)
BLSTM	1	0.0704	97.80
Derin BLSTM 1	2	0.0688	98.74
Derin BLSTM 2	3	0.0612	98.93
Derin BLSTM 3	4	0.0240	99.25
Derin BLSTM 4	5	0.0784	97.49
Derin BLSTM 5	6	0.0704	97.78

Gerçekleştirilen deneylerde katman sayısı arttıkça modelin başarısının arttığı gözlenmiştir. Ancak 5 ve 6 katmanlı modellerde katman sayısının artması ile kayıp

değerinin artım gösterdiği ve doğruluğun azaldığı gözlenmiştir. Bu azalmanın sebebini veri miktarı olarak yorumlamaktadır. Derlemdeki veri miktarı arttıkça çok katmanlı modellerin daha başarılı olacağını tahmin edilmektedir.

Tablo 6'da geliştirilen platformda kullanılan model ile diğer sözcük etiketleme modelleri karşılaştırılmaktadır.

Tablo 6 Sözcük etiketleme doğrulukları

Model	Doğruluk (Accuracy)
SVM with manual feature pattern (Gimenez ve Marquez)	97.16
MLP with word embeddings + CRF (Collobert ve ark)	97.29
LSTM (Huang ve ark.)	97,29
BLSTM (Huang ve ark.)	97,40

Tablo 6'da gösterilen Penn Treebank veri kümesinin Wall Street Journal bölümü derlemi üzerinde sözcük etiketleme modelleri son yıllarda yaygın olarak kullanılmaktadır. Gimenez ve Marquez yedi kelimelik bir pencerede elle tanımlanmış özelliklere dayanan SVM kullanarak 97,16 doğruluk sağlamışlardır. Özellik mühendisliğini azaltmak için, Collobert ve ark. yaptığı çalışmada çok katmanlı bir modele kelime gömme ve CRF'nin dahil edilmesinin de faydalı olduğunu kanıtlanmış ve 97,29 doğruluk sağlamışlardır. Huang ve ark. rastgele uzun bağlamı modellemek için kelime gömme ve kelime düzeyinde özellikleri bir araya getirmişlerdir. LSTM kullanarak 97,29 ve çift yönlü LSTM kullanarak 97,40 doğruluk sağlamışlardır (Young, Hazarika, Poria, & Cambria, 2018).

Derin BLSTM 3 bölüm 4 de açıklanan platformda kullanılan modeldir. Derin BLSTM 3 modeli ile geliştirilen Türkçe sözcük etiketleme için başarılı sonuçlar raporlanmıştır. Derin BLSTM 3 modeli kullanılarak %99,25 doğruluk ile Türkçe için en iyi sonuçlar alınmıştır.

Türkçe sözcük etiketleme platformu kullanıcılara tahmin edilen etiketleri düzeltme imkânı verilerek sistemin geliştirilmesine dâhil olmaları sağlanmıştır. Derin öğrenme kullanarak doğal dil işlemeyi gerçekleştiren ve kullanıcıları da sürece dâhil ederek doğal kavramının sürekliliğinin sağlayan Türk dilinde ilk platform olduğu düşünülmektedir.

Veri kümesi üzerinde yapılan otomatik etiketleme işlemenin başarılı bir şekilde yapılmadığının ve iyileştirilmesi gerektiği gözlemlenmiştir. Bilinmeyen sözcük türlerinin de derleme eklenmesinin başarılı sonuçlar elde edilmesine olumlu etkisi olacaktır. Sözcük

etiketlemesinin daha başarılı bir şekilde yapılması için veri üzerindeki etiketlerin doğruluğunun arttırılması ve modelin geliştirilmesi gerektiği düşünülmektedir.

Veri kümesi 34 konu ile sınırlandırılmıştır. Farklı konuların eklenmesi veri çeşitliliğinin artmasını sağlayacaktır. Bu şekilde elde edilen sonuçların doğal dile yaklaşacağı düşünülmektedir.

İleriki çalışmalarda, bu tez çalışmasında geliştirilen modelde kullanılan doğru etiketlenmiş eğitim verisinin miktarı ve donanımsal özellikler arttırılarak Türkçe için başarılı ve hızlı sonuçlar elde edilmesi hedeflenmektedir.

KAYNAKLAR

Aksan, M., & Aksan, Y. (2018). Linguistic corpora : A view from Turkish. *Studies in Turkish Language Processing* (s. 301-327). içinde Springer Verlag.

Aksan, Y., & Aksan, M. (2018). *Studies in Turkish Language Processing.* Springer Verlag.

Aksan, Y., & Yaldır, Y. (2012). A corpus-based word frequency list of Turkish: Evidence from the Turkish National Corpus. *15 th International Conference on Turkish Linguistics.*

Baker, P., Hardie, A., & McEnery, T. (2006). *A Glossary of Corpus Linguistics.* Edinburgh University Press.

Blum, A., & Langley, P. (1997). Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 245-271.

Bobrow, D. G. (1967). Natural Language Input for a Computer Problem Solving System. *In Semantic Information Processing*, 133-215.

Brill, E. (1992). A Simple Rule-Based Part of Speech Tagger. *ANLC '92 Proceedings of the third conference on Applied natural language processing.* Trento.

Buchanan, B. (2006). A (very) Brief History of Artificial Intelligence. *AI Magazine*, 53-60.

Cambria, E., & White, B. (2014). Jumping NLP curves: A review of natural language processing research. *IEEE Computational Intelligence Magazine, 9*(2), 48-57.

Chomsky, N. (1986). *Knowledge of Language: Its Nature, Origin, and Use.* New York.

Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *Neural and Evolutionary Computing*.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P. (2011). Natural Language Processing (Almost) from Scratch. *Journal of Machine Learning Research, 12*, 2493-2537.

Corpus Types. (2019, 05 23). www.sketchengine.eu: https://www.sketchengine.eu/user-guide/user-manual/corpora/corpus-types/ adresinden alındı

Cortes, C., & Vapnik, V. (1995). Support-Vector Networks. *Machine Learning, 20*(3), 273-297.

Çöltekin, Ç. (2014). A Set of Open Source Tools for Turkish Natural Language Processing. *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, (s. 1079–1086).

Davies, M. (2009). The 385+ million word Corpus of Contemporary American English (1990-2008+): Design, architecture, and linguistic insights. *International Journal of Corpus Linguistics*, 159-190.

Deng, L., & Yu, D. (2014). Deep Learning: Methods and Applications. *Foundations and Trends in Signal Processing* (Cilt 7, s. 197-387). içinde

Dobbin, K. K., & Simon, R. M. (2011). Optimally splitting cases for training and testing high dimensional classifiers. *Dobbin and Simon BMC Medical Genomics, 4*(31).

Dunne, R. A., & Campbell, N. A. (1997). On The Pairing Of The Softmax Activation And Cross-Entropy Penalty Functions And The Derivation Of The Softmax Activation Function. *Conf. on Neural Networks*, (s. 181-185). Melb.

Elman, J. L. (1990). Finding Structure in Time. *Cognitive science*, 179-211.

Gimenez, J., & Marquez, L. (2004). Svmtool: A general pos tagger generator based on support vector machines. *In Proceedings of the 4th International Conference on Language Resources and Evaluation.*

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning.* MIT Press.

Granger, S. (2003). The International Corpus of Learner English: A New Resource for Foreign Language Learning and Teaching and Second Language Acquisition Research. *TESOL Q*, 538-546.

Graves, A., & Schmidhuberab, J. (2005). Framewise Phoneme Classification with Bidirectional LSTM and Other Neural Network Architectures. *Neural Networks*, 602-610.

Haugeland, J. (1985). *Artificial Intelligence.* MIT Press.

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation, 9*(8), 1735-1780.

Ivakhnenko, A., & Lapa, V. (1965). Cybernetic predicting devices.

Kilimci, A., & Can, C. (2009). TICLE: Uluslararası Türk Öğrenci İngilizcesi Derlemi. *Proceedings of the national linguistics conference*, (s. 1-11). Van.

Kingma, D. P., & Ba, J. L. (2015). Adam: A Method for Stochastic Optimization. *3rd International Conference for Learning Representations.*

Köksal, A. (1976). *A first approach to a computerized model for the automatic morphological.* Ankara: Hacettepe University.

Köksal, A. (1976). A first approach to a computerized model for the automatic morphological analysis of Turkish.

Lally, A., Prager, J., McCord, M., Boguraev, B., Patwardhan, S., Fan, J., . . . Carroll, J. (2012). Question analysis: how Watson reads a clue. *IBM Journal of Research and Development*.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *521*(7553), 436-444.

McCarthy, j., Minsky, M., Rochester, N., & Shannon, C. (1955). A proposal for The Darthmouth Summer Research Project on Artificial Intelligence., (s. Dartmouth Artificial Intelligence (AI) Conference).

Mikolov, T., Karafiat, M., Burget, L., Cernocky, J., & Khudanpur, S. (2010). Recurrent Neural Network Based Language Model. *İnterspeech*, *2*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., & Dean, J. (2013). Distributed Representations of Words and Phrases and their Compositionality. *Proceedings of the 26th International Conference on Neural Information Processing Systems*, (s. 3111-3119). Lake Tahoe, Nevada.

Minsky, M. (1952). *A neural-analogue calculator based upon a probability model of reinforcement.* Technical report Harvard University Psychological Laboratories.

Ozturk, S., Sankur, B., Gungor, T., & Yilmaz, M. B. (2014). Türkçe Etiketli Metin Derlemi (Turkish Labeled Text Corpus). *2014 22nd Signal Processing and Communications Applications Conference (SIU).*

Papert, S., & Minsky, M. (1988). *Perceptrons: an introduction to computational geometry.* MIT Press.

Pitts, W., & McCulloch, W. (1943). A logical calculus of ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 115-133.

Poole, D., & Mackworth, A. (2019, 05 24). *Foundations of Computational Agents.* Artifical Intelligence 2E: https://www.artint.info adresinden alındı

Rissanen, M., Kytö, M., Tarkka, L., Kilpiö, M., Nevanlinna, S., Pahta, P., . . . Brunberg, H. (1991). *The Helsinki corpus of english texts.* Helsinki: The University of Helsinki.

Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review*, 386-408.

Ruhi, Ş., Tuğa, B., Hatipoğlu, Ç., Güler, H., Acar, M., Eryılmaz, K., . . . Karadaş, D. (2010). Sustaining a Corpus for Spoken Turkish Discourse: Accessibility and Corpus Management Issues. *Proceedings of the workshop on language resources: from storyboard to sustainability and LR lifecycle management*, (s. 44-48). Valetta.

Russell, B., & Whitehead, A. (1910). *Principia mathematica.* Cambridge University Press.

Samuel , A. (1959). Some studies in machine learning using the game of checkers. *BM Journal on Research and Development*, 210-229.

Say, B., & Özge, U. (2004). Development of a Corpus Workbench for the METU Turkish Corpus. *Proceedings of LREC*, 223-225.

Schmidhuber, J. (2015). Deep Learning in Neural Networks: An Overview. *61*, 85-117.

Schuster, M., & K. Paliwal, K. (1997). Bidirectional Recurrent Neural Networks. *IEEE Transactions on Signal Processing, 45*(11), 2673-2681.

Schuster, M., & K. Paliwal, K. (1997). Bidirectional Recurrent Neural Networks. *IEEE TRANSACTIONS ON SIGNAL PROCESSING, 45*(11), 2673 - 2681.

Schütze, H., & Manning, C. D. (1999). *Foundation of Statistical Natural Language Processing.*

Song, H., & Lee, S.-Y. (2013). Hierarchical Representation Using NMF. *International Conference on Neural Information Processing 2013: Neural Information Processing*, (s. 466-473).

Sundermeyer, M., Alkhouli, T., Wuebker, J., & Ney, H. (2014). Translation Modeling with Bidirectional Recurrent Neural Networks. *EMNLP Conference*, (s. 14-25). Doha, Qatar.

Sundermeyer, M., Ney, H., & Schluter, R. (2015). From Feedforward to Recurrent LSTM Neural Networks for Language Modeling. *IEEE/ACM Transactions on Audio, Speech and Language Processing, 23*(3), 517-529.

Sundermeyer, M., Schlüter, R., & Ney, H. (2012). LSTM Neural Networks for Language Modeling. *Interspeech.* USA.

Şeker, A., Dirib, B., & Balık, H. H. (2017). Derin Öğrenme Yöntemleri ve Uygulamaları Hakkında Bir İnceleme. *Gazi Journal of Engineering Sciences (GJES), 3*(3), 47-64.

Tang, D., Qin, B., & Liu, T. (2015). Document Modeling with Gated Recurrent Neural Network for Sentiment Classification. *EMNLP*, (s. 1422–1432).

Turing, A. (1950). Computing Machinery and Intelligence. *Mind*, 433-460.

Universal Dependencies. (2019, 05 25). (Universal Dependencies) https://universaldependencies.org/ adresinden alındı

Voskoglou, C. (2017, 5 4). *What is the best programming language for Machine Learning*. 2019 tarihinde developereconomics: https://www.developereconomics.com/best-machine-learning-language adresinden alındı

Wang, H. (1960). Toward mechanical mathematics. *IBM Journal of Research and Development*, 2-22.

Wang, P., Qian, Y., K. Soong, F., He, L., & Zhao, H. (2015, 10 21). Part-of-Speech Tagging with Bidirectional Long Short-Term Memory Recurrent Neural Network. arXiv:1510.06168v1 [cs.CL]. adresinden alındı

Warren , D., & Pereira , F. (1982). An efficient easily adaptable system for interpreting natural language queries. *Computational Linguistics*, 110-122.

Winograd, T. (1972). *Understanding natural language.* Academic Press.

Yao, K., Zweig, G., Hwang, M.-Y., Shi, Y., & Yu, D. (2013). Recurrent Neural Networks for Language Understanding. *Interspeech*, (s. 2524–2528).

Young, T., Hazarika, D., Poria, S., & Cambria, E. (2018). Recent Trends in Deep Learning Based Natural Language Processing. *IEEE Computational Intelligence Magazine*, 55-75.

ÖZGEÇMİŞ

Kişisel Bilgiler

Adı, soyadı : Barış BABÜROĞLU

Uyruğu : T.C.

Doğum tarihi ve yeri : 13.10.1992, Gaziantep

Medeni hali : Evli

Telefon : 0 (542) 726 86 52

e-posta : barisbaburoglu@gmail.com

Eğitim

Derece	Eğitim Birimi	Mezuniyet tarihi
Yüksek lisans	KSÜ /Enformatik Anabilim Dalı	2016- 2019
Lisans	Fırat Üniversitesi / Bilgisayar Müh. Bölümü	2010 -2014

İş Deneyimi

Yıl	Yer	Görev
2015-	Gaziantep Büyükşehir Belediyesi	Mühendis

Yabancı Dil

İngilizce

Yayınlar

1. Babüroğlu, B., Tekerek, A., & Tekerek, M., (2019). Türkçe için Derin Öğrenme Tabanlı Doğal Dil İşleme Modeli Geliştirilmesi, 13. Bilgisayar ve Öğretim Teknolojileri Sempozyumu, Kırşehir, 2-4 Mayıs 2019

İlgi Alanları

Robotik, Yapay Zekâ, Doğal Dil İşleme