T.C.
ERZİNCAN BİNALİ YILDIRIM ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS TEZİ

DOĞAL DİL İŞLEME İLE TERÖR TEHDİT UNSURLARININ TESPİT EDİLMESİ

Ahmet GÜLER

Danışman: Dr. Öğr. Üyesi İsmail AKGÜL

YAPAY ZEKA VE ROBOTİK
ANABİLİM DALI

ERZİNCAN
2023

Kabul ve Onay Sayfası

Dr. Öğr. Üyesi İsmail AKGÜL danışmanlığında, Ahmet GÜLER tarafından hazırlanan bu çalışma 17/07/2023 tarihinde aşağıdaki jüri tarafından Yapay Zeka ve Robotik Anabilim Dalı'nda Yüksek Lisans Tezi olarak oy birliği ile kabul edilmiştir.

Başkan:	Doç. Dr. Durmuş ÖZDEMİR	İmza:
Üye :	Dr. Öğr. Üyesi İsmail AKGÜL	İmza:
Üye :	Dr. Öğr. Üyesi Volkan KAYA	İmza:

Yukarıdaki sonuç Enstitü Yönetim Kurulunun …. / …. / 20.... tarih ve …../…….. sayılı kararı ile onaylanmıştır.

Prof. Dr. Bülent ÇAĞLAR

Enstitü Müdürü

Not: Bu tezde kullanılan özgün ve başka kaynaklardan yapılan bildirişlerin, şekil ve tabloların kaynak olarak kullanımı, 5846 sayılı Fikir ve Sanat Eserleri Kanunundaki hükümlere tabidir.

Bilimsel Etiğe Uygunluk Sayfası

“Doğal Dil İşleme İle Terör Tehdit Unsurlarının Tespit Edilmesi” isimli “Yüksek Lisans” tezim tarafımca intihal tespit programı ile incelenmiştir. Buna göre tezimde bilimsel etik ihlali ve intihal olarak nitelendirilebilecek herhangi bir durum olmadığını taahhüt ederim.

Bu çalışmadaki tüm bilgilerin, akademik ve etik kurallara uygun bir biçimde elde edildiğini; aynı zamanda bu kural ve davranışların gerektirdiği gibi, bu çalışmanın özünde olmayan tüm materyal ve sonuçları tam olarak aktardığımı ve referans gösterdiğimi beyan ederim. 17/07/2023

(İmza)

Ahmet GÜLER

ÖZET

Yüksek Lisans Tezi

DOĞAL DİL İŞLEME İLE TERÖR TEHDİT UNSURLARININ TESPİT EDİLMESİ

Ahmet GÜLER

Erzincan Binali Yıldırım Üniversitesi
Fen Bilimleri Enstitüsü
Yapay Zeka ve Robotik Anabilim Dalı

Danışman: Dr. Öğr. Üyesi İsmail AKGÜL

Günümüz dünyasında milyonlarca verinin anlaşılır hale getirilmesi önem kazanmıştır. Özellikle bu veriler kullanılarak suç teşkil eden konularda daha hızlı adımlar atılabilmesi için veri analizinin hızlı yapılabilmesi gerekmektedir. Bu kapsamda yapay zeka biliminin doğal dil işleme yöntemi ile gerçekleştirilen duygu analizi, olası can ve mal kayıplarının ortadan kaldırılmasına imkân sağlamaktadır. Ayrıca olası terör bölgelerinde bütün telsiz frekansları aynı anda dinlenerek terör örgütlerinin saldırıları doğal dil işleme yöntemleri ile analiz edilerek saldırı gerçekleşmeden engellenebilir. Bu yüksek lisans tez çalışmasında, terör tehdit unsurlarının tespiti için sanal ortamdaki metin, ses ve görüntü verilerinin analizinde yapay zeka biliminin doğal dil işleme yöntemleri kullanılmıştır. Böylelikle terör bölgelerinde terör unsurlarının görüşmelerinin analizi yapılarak, özellikle kolluk kuvvetlerinin sağlıklı müdahalesi ve can güvenliğinin sağlanması amaçlanmıştır. Tez çalışmasında doğal dil işleme bilimi ile ilgili kavramsal yapı ve doğal dil işleme yöntemleri detaylı bir şekilde ele alınmıştır. Ele alınan yöntemler ile modeller oluşturulup sonuçlar sunulmuştur.

2023, 53 Sayfa

Anahtar Kelimeler: Derin öğrenme, doğal dil işleme, veri ön işleme, öznitelik çıkarımı, sınıflandırma yöntemleri, terör-tehdit unsurları.

ABSTRACT

Master Thesis

DETECTING TERROR THREAT ELEMENTS BY NATURAL LANGUAGE PROCESSING

Ahmet GÜLER

Erzincan Binali Yıldırım University
Graduate School of Natural and Applied Sciences
Department of Artificial Intelligence and Robotics Department

Supervisor: Asst. Prof. Dr. İsmail AKGÜL

In today's world, it has become important to make millions of data understandable. In order to take faster steps in criminal matters, especially by using these data, data analysis should be done quickly. In this context, sentiment analysis performed with the natural language processing method of artificial intelligence enables the elimination of possible loss of life and property. In addition, by listening to all radio frequencies at the same time in possible terrorist areas, the attacks of terrorist organizations can be analyzed with natural language processing methods, so that the attack can be prevented before it takes place. In this master's thesis study, natural language processing methods of artificial intelligence were used in the analysis of text, audio, and image data in the virtual environment for the detection of terrorist threat elements. In this way, it is aimed to ensure the healthy intervention of law enforcement officers and the security of life by analyzing the talks of terrorist elements in terror zones. In the thesis study, the conceptual structure and natural language processing methods related to natural language processing are mentioned in detail. Models were created with the mentioned methods and the results were presented.

2023, 53 Pages

Keywords: Deep learning, natural language processing, data preprocessing, feature extraction, classification methods, terror-threat elements.

TEŞEKKÜR

Yüksek lisans süresince fedakârlığını, tecrübesini ve bilgilerini esirgemeyen, sürecin her anında görüşlerine başvurduğum ve sevgisini her an hissettiğim danışmanım saygıdeğer Dr. Öğr. Üyesi İsmail AKGÜL'e en kalbi teşekkürlerimi sunarım.

Bu süreçte de hayatım boyunca olduğu gibi sabırla çalışmalarımı dinleyen ve benim mutluluğum ile mutlu olan eşim Zeynep GÜLER'e, çalışmalarım sırasında bazen vakit ayıramadığım oğlum Ömer Ali GÜLER ve kızım Fatıma Reyyan GÜLER'e sonsuz teşekkürlerimi sunarım.

Ahmet GÜLER
Temmuz, 2023

İÇİNDEKİLER

Sayfa

ŞEKİLLER LİSTESİ

Sayfa

TABLOLAR LİSTESİ

Sayfa

SİMGELER ve KISALTMALAR

Simgeler

%	Yüzde
E	Regresyon Sabiti

Kısaltmalar

BERT	Bidirectional Encoder Representations from Transformers
CBOW	Continuous Bag of Words
DDİ	Doğal Dil İşleme
ELMo	Embeddings from Language Model
GloVe	Global Vectors for Word Representation
GRU	Gated Recurrent Units
IDF	Inverse Document Frequency
LSTM	Long Short-Term Memory
NLP	Natural Language Processing
Spark NLP	State-of-the-Art Natural Language Processing
TF	Term Frequency
USE	Universal Sentence Encoder

1. GİRİŞ

Yapay zeka, insana ait olan görebilme, duyabilme, konuşabilme, dokunabilme gibi özelliklerin bilgisayar sistemine kazandırılabilme sürecidir. Bu özelliklerin kazandırma aşamasından sonra nihai amaç, bilgiye dönüşen veriden doğru karar verebilmektir. Bu sayede insani ihtiyaçlar olan yemek ve içmek gibi istekler olmadan, bilgisayarların makineleri insan zekası gibi yönetebilmesi hedeflenmiştir. Sonucunda beyni bilgisayar olan akıllı makineler üretilmiş olacaktır (Demir, 2012).

Günlük hayatta birçok alanda yapay zeka hayatın içerisinde yer edinmiştir. Yapay zeka, ses işleme (ses tanıma, sesli asistan, sesli yanıt, konuşmadan metin sentezi, metinden konuşma sentezi), metin işleme (çeviri servisleri, çevrimiçi sohbet ve asistan, sosyal medya analitiği ve duygu analizi, kişiye özgü yazı düzenleme ve öneri), görüntü işleme (güvenlik, yüz tanıma ve gözetleme, sosyal ortamda fotoğraf bulma), sağlık verilerinin analizi ve tedavi planlaması (tanı koyma ve tedavi planlama sürecinde sağlık personeline yardımcı olan uygulamalar), otonom sürüş sistemleri (otonom araçlarda karar destek sistemleri), büyük veri analitiği (büyük veri analizi ile davranış analizi), veri işleme (öneri sistemleri, ilan öneri, müzik öneri, müşteri deneyim ve müşteri için akıllı kampanya öneri), tarım ve hayvancılıkta akıllı uygulamalar (görüntü işleme temelli hassas tarım uygulamaları, hassas hayvansal üretim), siber güvenlik (siber saldırı tespit ve engelleme için uzman sistemi, kötü amaçlı yazılım analizi) gibi alanlarda kullanımını her geçen gün arttırmıştır (Türkiye Cumhuriyeti Cumhurbaşkanlığı, 2022).

İnsanlığın var oluşundan günümüze kadar kullandığı konuşma dilinin kendine ait özellikleri ve yapısı vardır. Yapay zekanın bir alt dalı olan doğal dil işleme, dünya dillerinin özelliklerinin çözümlemesini ve duygu analizini yapan bir tekniktir. Doğal dil işleme insan ve makine arasındaki dilin oluşturulmasında en önemli yapıtaşıdır. Bu sayede dijital ortamda metinsel verilerin istenilen dile dönüştürülmesini, sorulan sorulara mantıklı cevap verilebilmesini, dijital metinlerin sentezinin ve özetinin yapılabilmesini, sesli komutlar ile makinelerin yönlendirilebilmesini sağlar (Ballı, 2021). Günümüz teknolojisinde halen istenilen seviyeye gelmeyen doğal dil işleme tekniği gelişimine devam etmektedir.

Doğal dil işleme biliminin gelişme aşamasında birçok yöntem kullanılmıştır. Bu yöntemlerin sonuç değerleri, model eğitim süreleri, verilerin sayısı ve çeşitliliği doğal dil işleme biliminin kendini ispatlamasında önemli parametrelerdir. Son yıllarda en çok merak uyandıran yöntemlerin başında derin öğrenme yöntemi gelmektedir. Derin öğrenme ile dil model çözümlenmesi, ses analizi, karar verme sistemleri, görsel veriler, çok giriş ve çıkışlı modeller oluşturabilmek mümkündür (Deng ve Yu, 2014). Derin öğrenme dil model çözümlemesi, herhangi bir dil üzerinden duygu ifadelerini anlamamıza yardım eder. Böylelikle çözümleme yapılan dil üzerinde niyet analizi ve duygu sınıflandırması yapılabilir. Ses insanların duygu yükünü, kelimelerden daha fazla taşıyan duygu yapı taşlarından biridir. Derin öğrenme yöntemiyle ses analizi yapılarak kişinin sahip olduğu duygu fikrine ulaşmak mümkündür. Böylece derin öğrenmeyle sınıflandırma yapılarak bir karara varılabilir. Ayrıca CNN derin öğrenme yöntemiyle görsel verilerin sınıflandırması, RNN yöntemiyle görüntüden metin elde edilebilmesi mümkündür. Kişinin sahip olduğu sinirli, mutlu, korkulu, öfkeli gibi birçok çıkışa sahip olan sınıflandırma işlemi, yine derin öğrenme yöntemiyle olanaklı hale gelir (Küçük ve Arıcı, 2018).

Bu tez çalışmasının ilk aşamasında yapay zeka biliminin alt dallarından biri olan doğal dil işleme yönteminin tanıtımı, insan ve bilgisayar arasındaki dilin oluşturulması, metinsel verilerin temizlenmesi, öznitelik çıkarılması ve sınıflandırılması konularında bilgiler sunulmuştur. İkinci aşamasında yapay zeka modelinde kullanılan materyallerin içeriği ve kullanılan yöntemlerin açıklamaları bulunmaktadır. Üçüncü aşamasında ise oluşturulan modellerin eğitim sonuçları ile karşılaştırılmaları yapılmıştır. Oluşturulan her modelde terör tehdit unsurlarının tespit edilmesi hedeflenmiştir.

2. KAYNAK ÖZETLERİ

2.1. Doğal Dil İşleme ile Duygu Analizi Çalışmaları

Seker (2016), çalışmasında makine öğrenmesi yöntemiyle duygu analizi yapmıştır. Doğal dil işleme öncesinde öznitelik çıkarımı için TF-IDF yönetimini kullanarak kelimelerin değer sıralamasını ortaya çıkarmıştır. Öznitelik işleminden sonra Linear Regression makine öğrenme yöntemini kullanarak Türkçe verilerde %83, İngilizce verilerde %74 doğruluk oranında duygu sınıflandırması yapmıştır.

Tuzcu (2020), kullanıcı yorumlarının duygu sınıflandırmasını yapmıştır. Çalışmasında Destek Vektör Makinesi, Naive Bayes ve Lojistik Regresyon yöntemlerini kullanmıştır. Destek Vektör Makinesi yöntemiyle %80,93, Naive Bayes yöntemiyle %77,57 ve Lojistik Regresyon yöntemiyle %84,07 doğruluk oranına ulaşmıştır.

İlhan ve Sağaltıcı (2020), çalışmalarında Twitter üzerinden alınan yorumlar üzerinden duygu analizi yapmıştır. Modellerini Destek Vektör Makinesi, Naive Bayes ve Vader üzerinden çalıştırmış olup, en yüksek doğruluk oranına %64 ile Destek Vektör Makinesi ile ulaşmışlardır.

Yilmaz vd. (2021), Türkçe metinler üzerinden duygu analizini derin öğrenme ile gerçekleştirmişlerdir. Kullanılan verilerinin çeşitliliği modelin güvenilir olmasında büyük bir öneme sahip olduğundan, büyük verilerde derin öğrenmenin başarısından bahsedilmiştir. Derin öğrenmenin, makine öğrenmesi gibi diğer yöntemlere göre büyük verilerde modelin doğrulunda %40 iyileştirme sağladığı bilgisine ulaşılmıştır.

Yılmaz ve Orman (2021), Covid 19 sürecinde Twitter üzerinden alınan yorumlar ile duygu analizi gerçekleştirmişlerdir. Destek Vektör Makinesi ve Lojistik Regresyon yöntemleri kullanarak sırası ile %75 ve %74,75 doğruluk oranına ulaşılmıştır. BERT ile yapılan modelde ise %89 doğruluğa ulaşıldığı görülmüştür. Duygu analizini cümle yapısından kelime yapısına indirgeyip yapılan eğitimlerde, modelin doğruluğunun %98,13 oranına ulaştığı ifade edilmiştir.

Ekim ve İnner (2021), doğal dil işleme ile yapılan birçok çalışmanın incelemelerini yapmışlardır. Bu çalışmalarda makine öğrenme yöntemleri, derin öğrenme yöntemleri

ve kelime/cümle vektör oluşturma yöntemleri bulunmaktadır. 2017 yılından beri doğal dil işleme ile duygu analizi ve fikir madenciliği çalışmalarında azımsanmayacak kadar model oluşturulduğunu göstermişlerdir.

Yurt (2015), sanal ortamdan çekilen 2000 yorumun duygu analizini yapmıştır. Yorumları pozitif ve negatif olmak üzere 2 grupta sınıflandırmıştır. Çalışmasında TF-IDF yöntemini kullanmış olup, yaklaşık %74'lük bir doğruluk oranını yakalamıştır.

Bostancı ve Albayrak (2021), doğal dil işleme ile duygu analizi yaparak kişiye özel içerik önerme üzerine çalışma yapmışlardır. Verileri sanal ortamdaki yorumlardan alıp modellerini eğitmişlerdir. Eğitim sonucunda %83 doğruluk oranına ulaşıldığı ifade edilmiştir.

Polat ve Ağca (2022), çalışmalarında internette yapılan yorumları Random Forest yöntemi kullanılarak, olumlu ve olumsuz olarak iki sınıfa ayırma işlemini gerçekleştirmişlerdir. Yaklaşık %77 doğruluk oranını elde etmişlerdir.

Atlı ve İlhan (2021), doğal dil işleme ile duygu sözlüğü oluşturulması konusunda çalışma yapmışlardır. Çalışmalarında, makine öğrenme yöntemlerini kullanarak 38 farklı duygunun çıkartılabileceğini göstermişlerdir.

2.2. Doğal Dil İşleme ile İlgili Diğer Çalışmalar

Albayrak (2020), çalışmasında doğal dil işleme ile lisansüstü ders müfredat ve içeriklerinin hazırlanmasını ele almıştır. Veri temizleme, veri ön işleme yapıldıktan sonra TF-IDF sınıflandırma yöntemini kullanmıştır. Çalışmasının sonunda hafta dağılımı yaparak, her hafta işlenmesi gereken dersleri ve sınav günlerini gösteren tablo şemasını elde etmiştir.

Toğaçar vd. (2021), internet ortamında çıkan yalan haberlerin tespiti için doğal dil işleme tekniğini kullanmışlardır. 3171 gerçek ve 3164 sahte haberi modeli eğitmek için kullanmışlardır. LSTM yöntemi kullanılarak oluşturulan modelde %91,48 gibi çok yüksek bir başarı oranına ulaşmışlardır.

Canım (2019), eski dillerde kullanılan kelimelerin anlamsal yakınlıklarını gösteren bir çalışma gerçekleştirmiştir. Doğal dil işlemenin kelime vektörü oluşturma yöntemi ile model eğitimi gerçekleştirilmiştir. Modelde eski Türkçede kullanılan günümüzde kullanılmayan kelimeler veri setine dâhil edilmiştir. Çalışmada CBOW ve Skip-gram yöntemleri ile kelime vektörleri oluşturulmuş olup, CBOW ile daha yüksek performanslı sonuçlar elde edilmiştir.

Kontuk ve Turan (2020), yaptıkları çalışmalarda doğal dil işleme bilimi ile haber sitelerindeki yorumları yaş gruplarına göre sınıflandırmışlardır. Toplamda 3925 yorumdan oluşan veri setinin, %70'ini eğitim ve %30'unu test olarak ayırma işlemi yapmışlardır. Terim frekansı yöntemiyle oluşturulan modelde, %70 doğruluk oranına ulaşılmıştır.

Dayan ve Yılmaz (2022), doğal dil işleme ve derin öğrenme yöntemleriyle sesleri sınıflandırma ve ses üretme üzerine çalışma yapmışlardır. CNN, LSTM-NLP ve YSA yöntemlerini kullanarak çeşitli sesleri Kaggle internet sitesi üzerinden alarak eğitim modelleri oluşturmuşlardır. Ses sınıflandırmasında en yüksek başarıya LSTM-NLP ile %91 doğruluk oranı ile yaklaşmışlardır. Aynı zamanda makinelerin kendi aralarında haberleşmesini ve dış dünyadaki seslerin birbirine benzeşmesini sağlayan bir model ortaya çıkarmışlardır.

Küçük ve Arıcı (2018), doğal dil biliminde derin öğrenmenin hızla artan önemi hakkında çalışma yapmışlardır. Metin sınıflandırma, bilgi çıkarımı, soru cevaplama, metin ayrıştırma gibi birçok alanda doğal dil işleme biliminin önemini belirtmişlerdir.

Aksoy (2021), çalışmasında doğal dil işleme ile açık uçlu sınavların değerlendirilmesini ele almıştır. Çalışmasında coğrafya dersi için kullandığı KNN algoritmasıyla %97 doğruluk oranına ulaşılmıştır. Bu sonucun dersler içerisinde en yüksek doğruluk oranı olduğu gösterilmiştir.

Oflazer (2016), doğal dil işleme bilimi ile Türkçe dilleri üzerinde çalışma yapmıştır. Türkçe dilinin cümle yapısını, dil modelini ve anlam çıkartma işlemlerini kelime/vektör yöntemini kullanarak uygulamayı başarmıştır.

Delibaş (2008), çalışmasında doğal dil işleme ile Türkçe dilinde yazım hataları bulma işlemini ele almıştır. Kök bulma yöntemini kullanarak model oluşturulmuştur. Eğitim sonucunda %92 doğruluk oranına ulaşılmıştır.

3. KURAMSAL TEMELLER

3.1. Doğal Dil İşleme Kavramı

İnsanoğlu günümüze kadar kendi aralarında anlaşıp duygu aktarımı yapabilmek için çeşitli diller kullanmıştır. Yapay zeka biliminin çok önemli bir alt kümesi olan doğal dil işleme tekniği de insanlar ile makineler arasında bir köprü vazifesi almıştır. Doğal dil işleme tekniği birçok alanda kullanılmaya başlamış ve halen geliştirilmeye devam etmektedir. Günlük hayatımızda da yer alan doğal dil işleme uygulamaları; makine çevirisi, metin özetleme, soru yanıtlama, duygu analizi, sanal asistanlar, dil ve varlık tanıma, sözcüklerdeki anlam belirsizlik tespiti, intihal tespiti, ses bilimi gibi birçok alanda kullanılmaya başlanıp geliştirme süreçleriyle devam etmiştir (Özmutlu, 2021).

Doğal dil işleme yönteminin adımları Şekil 3.1'de ifade edildiği basamaklardan oluşmaktadır. Şekilde de görüldüğü gibi metin olarak sisteme giren ham veriler belirli işlemlerden geçirilip sayısal verilere dönüştürülerek eğitime uygun hale getirilir. Eğitime uygun hale getirilen verilerle model eğitimleri gerçekleştirilir ve çıkış tahmini yapılır. Yani giriş verisi, veri ön işleme, kök bulma, öznitelik çıkarımı gibi süreçlerinden geçirildikten sonra eğitim sürecine tabi tutulur ve çıkış tahmini elde edilir.

Şekil 3.1. Doğal dil işleme adımları

3.2. Veri Ön İşleme

Doğal dil işleme sürecinin metinsel verileri anlaşılır hale getirebilmesinin ilk adımı veri ön işlemedir. Veri ön işleme karakterlerin sayısal hale dönüştürüp makine diline çevrildiği sürece denir (Ballı, 2021).

İnternet ortamından çekilen verilerin ya da hazır alınan veri setlerinin büyüklüğü milyonlarca kelimeden oluşabilmektedir. Bu kadar veri kümesinin bulunduğu bir dünya üzerinde yapılacak doğal dil işleme yöntemlerinden sağlıklı bir sonuç alınabilmesi, verilere ön işleme uygulanması sonucunda sağlıklı bir sonuç veya tahminde bulunmakla mümkün olacaktır. Aksi durumda oluşturulan modellerin doğruluk oranları çok düşük olup istenilen başarıyı sağlayamayacaktır.

3.2.1. Gürültülü verilerin temizlenmesi

Makine ve insan arasındaki iletişimi kurarken duygu ifadesi vermeyen ve eğitimde başarıyı düşüren karmaşaya gürültü denir (Ballı, 2021). Bu adımda, bu gürültü unsurlarının temizlenmesi amaçlanır. Kelimeler arasında anlam ifade etmeyen sayılar, bağlaçlar, noktalama işaretleri veya simgeler gürültü olarak adlandırılır. Konuşmalar veya web ortamından elde edilen veri setlerinin eğitilebilmesi ve sağlıklı sonuç alınabilmesi için öncelikle bu gürültü kaynaklarının temizlenmesi gerekmektedir. Bu sayede verilerdeki eksiklikler giderilir ve verilerdeki anlam bütünlüğü sağlanmış olur (Toprak, 2019).

Tüm kelimeleri büyük ya da küçük harfe dönüştürerek, metindeki noktalama işaretlerini, sayıları ve az geçen kelimeleri silerek gürültülü verilerin temizlenmesi gerekir. Gürültülü verilerin temizlenmesi için kullanılan yöntemlerin başlıcaları aşağıda verilmiştir.

3.2.1.1. Büyük-Küçük harf dönüşümü

Normalizasyon çalışmalarının ilk adımı büyük-küçük harf düzensizliğinin ortadan kaldırılmasıdır (Yılmaz ve Yumuşak, 2021). Metin içerisinde aynı anlamı veya duyguyu ifade eden kelimelerin modelin eğitimi aşamasında gürültü oluşturmaması için bütün kelimeleri küçük harfe veya bütün kelimeleri büyük harfe dönüştürerek, harfin küçüklük veya büyüklüğünden kaynaklanan gürültü ortadan kaldırılmış olur. Böylelikle ön işleme esnasında kelimelerin sayısal değerlere dönüşümü esnasında, aynı anlam taşıyan kelimelerin farklı değerler alma sorunu ortadan kaldırılmış olur (Taşkıran, 2021).

Genellikle bu veri temizleme yönteminde, bir zorunluluk olmamakla birlikte büyük harflerin küçük harflere dönüştürülmesi uygun bulunur. Büyük harflerin kullanımının önemli olmadığı durumlarda, kelimelerin birbirine ortaklaşmasını daha rahat yakalayabilmek için karakter dizisindeki tüm harfler küçük harfe çevrilir.

3.2.1.2. Noktalama işaretlerinin silinmesi

Dijital dünyada bulunan metinlerde noktalama işaretlerinin yanlış olma ve anlamı bozma olasılığı vardır (Adalı, 2012). Ön işleme esnasında noktalama işaretlerinin silinmesi, gereksiz vektör değerlerinden sistemi arındırmış olur. Doğal dil işleme yönteminde duygu tahmini yaparken noktalama işaretleri anlam içermediği ve gürültü oluşturduğu için bu gürültü kaynaklarının temizlenmeden eğitim yapılması tahmin sonucunu etkileyecektir. Bu yüzden metin içerisindeki noktalama işaretleri silinmelidir. Bu yöntem doğal dil işlemenin vazgeçilmez veri temizleme aşamasıdır (İlhan ve Sağaltıcı, 2020).

Metin içerisinde noktalama işaretlerinin bulunması, eğitimler sonucunda elde edilecek tahmin sonucunu etkileyen bir faktör olduğu için noktalama işaretlerinin silinmesi önemli bir adımdır. Noktalama işaretlerinin silinmesi ile metin içerisindeki gürültü ortadan kaldırılarak tutarsızlıklar giderilir. Noktalama işaretleri boşluk karakteri ile değiştirilerek kaldırılmış/silinmiş olur.

3.2.1.3. Sayıların silinmesi

Metin analizinde sayılar da kelimeler gibi anlam ifade etmediği için, metin içerisinden anlam çıkartırken sayılar gürültü kaynağı oluşturur. Modelin eğitiminden önce sayıların silinmesi gerekmektedir (Hark vd., 2017).

Metinde bulunan sayıların silinmesi de noktalama işaretlerinin silinmesi gibi gürültüyü ortadan kaldıran bir yöntemdir. Sayılar boşluk ile yer değiştirilerek gürültü giderilir.

3.2.1.4. Az geçen kelimelerin silinmesi

Duygu tahmini yapılacak metin içerisinde anlam ifade etmeyen ve az geçen birçok kelime vardır. Bu az geçen kelimeler modeli oluşturup eğitilirken duygu

sınıflandırmasında karmaşıklığa sebep olabilir. Ön işlemede bu kelimelerin metinden çıkartılması tahmin sonucunu iyileştirir (Toprak, 2019).

Veri seti içerisinde kelimeler belirli sıklıklarda yer alırlar. Her kelimelerin kullanım yoğunluğuna bağlı olarak az geçen kelimeler belirlenerek veri setinden çıkarılmalıdır. Bu sayede tüm veri seti düşünüldüğünde, anlam ifade etmeyen kelimeler çıkarılarak karmaşıklık önlenmiş ve gürültü giderilmiş olur.

3.3. Kelimelere Ayırma (Tokenization)

Metin verilerinin içerisinde bulunan cümleleri en küçük anlam ifade eden kelimelere ayırma işlemidir. Bu sayede anlam içeren kelimeleri bir sınıf içerisinde yorumlama yapılabilir.

Örneğin, "bugün gittiğim restoran çalışanlarının müşteriye yaklaşımından memnun kaldım", kelimelere ayırma işleminden sonra ["bugün", "gittiğim", "restoran", "çalışanlarının", "müşteriye", "yaklaşımından", "memnun", "kaldım"] halini alır (Ballı, 2021).

3.4. Kelime Kökünü Bulma (Stemming)

Konuşma dillerinde kök bir kelimenin türevlenmesiyle yeni kelimeler elde edilir. Kelime köküne inme yöntemiyle kelimelerin anlam içeren köklerinin elde edilmesi sağlanır. Bu durum eğitim sırasında arama aşamasını kısaltır.

Örneğin, "sevecen" kelimesini "sev" köküne indirerek anlam olarak "sevmek" fiili elde edilir (Khyani vd., 2020). Kelime kökünü bulma aşaması, Tablo 3.1'de ifade edildiği gibi kelimenin kökünü bularak ön işleme adımını gerçekleştirir. Tabloda verilen "silahı", "silahcı", silahım", "silahlar" ve "silahta" kelimelerinin hepsinin kökü, kelime kökünü bulma yöntemiyle "silah" olarak bulunmuştur.

Tablo 3.1. Silah kelimesinin kök örneği

Sıra No	Kelime	Kök
1	silahı	silah
2	silahcı	silah
3	silahım	silah
4	silahlar	silah
5	silahta	silah

3.5. Kelime Köküne İnme (Lemmatization)

Kelime köküne inme de kelime kökünü bulma ile aynı işlemi izler. Tek farkı kelimenin kökünü bulurken, bu kelimeyle eş anlamlı diğer kelimelere de anlam katar.

Örneğin, “arkadaş” ve “dost” kelimelerini birbiriyle özdeşleştirir (Khyani vd., 2020).

Özellikle doğal dil işleme bilimi ile sesli asistan sistemleri oluşturulmak isteniyorsa kelime köküne inme yönteminin kullanılması, sağlıklı bir model için gereklidir (Doğuç vd., 2020).

3.6. Öznitelik Çıkarımı (Feature Extraction)

Yapay zeka sistemleri için metinsel veriler anlam ifade etmeyebilir (Khalid vd., 2014). Öznitelik çıkarımı metin içerisindeki kelimeleri sınıflandırırken özelliklerine göre ayırma yöntemidir. Bu sayede gürültü seviyelerini indirerek kelimelerin özüne ulaşılmış olur. Öznitelik çıkarımı yapılırken temel olarak harf ve kelime sayısı, özel karakterleri ve sayıları yakalayarak sınıflandırma işlemi yapılır (Wang vd., 2020).

Boyut indirgeme konusuyla ilişkili olan öznitelik çıkarımı, makine öğrenimi, derin öğrenme, doğal dil işleme, veri madenciliği, örüntü tanıma ve görüntü işleme gibi birçok alanda kullanılan bir yöntemdir. Öznitelik çıkarımı, nesneyi tanımlayan özelliklerdir ve girdi verilerini kullanarak bu verilerden türetilmiş değerler (öznitelikler) oluşturur.

3.6.1. Count Vectorizer yöntemiyle öznitelik çıkarımı

Metinsel verilerin analizinde kelimelerin sayısına bakarak kelime frekansının çıkartılabildiği öznitelik çıkarım yöntemidir. Bu öznitelik çıkarımı, sayısal olarak metinde en çok bulunan kelimeleri ya da frekansı en çok olan kelimeleri en değerli olarak kabul eder. Tablo 3.2'de herhangi bir metin içerisinde yer alan kelimelerin frekans sayısını veren bir tablo örneği gösterilmiştir.

Tablo 3.2. Kelimelerin frekans sayısı

Kelime	saldırı	sevgi	araba	kış	nefret
Frekans	3	7	2	1	2

Tablo 3.2'de ifade edilen sayısal değeler göz önüne alındığında kelime sayım öznitelik yaklaşımına göre en değerli kelime "sevgi" kelimesidir (Aksoy, 2021).

3.6.2. Word Level yöntemiyle öznitelik çıkarımı

Python kütüphanesinde "Word Vector" ya da "Word Level" diye adlandırılan bu yöntem, metinsel verilerdeki kelimelerin her birine bir indeks numarası vererek aynı olan kelimelerin sayısını belirleme mantığına dayanır. Fazla geçen kelimelerin duygu anlamının doğruluğunu arttırarak diğer kelimeler ile de anlam benzerliğini gösteren bir öznitelik çıkarım yöntemidir **(**Python, 2022**)**.

Word Level, temelinde Count Vectorizer yöntemiyle aynı işlemi yapar, ancak aralarındaki en büyük fark TF-IDF Vectors yönteminde kelimelerin göreceli etkileri alınır. Önemli ve sık kullanılan kelimelerin göreceli etkileri daha fazlayken, önemli olmayan kelimelerin göreceli etki değeri küçüktür.

3.6.3. N-Gram Level yöntemiyle öznitelik çıkarımı

Bu yöntem, metni oluşturan kelimelerin veya kelimeyi oluşturan harflerin yan yana gelmesiyle oluşan örüntüyü inceleyerek, özellik çıkarılmasını sağlayan bir öznitelik çıkarma yöntemidir (Oğuzlar, 2011). Birlikte kullanılan kelimelerin kombinasyonunu gösterir. Belirtilen sayı kadar kelimeleri kombin eder. 1 kelimelik gruplar için unigram, 2 kelimelik gruplar için bigram, 3 kelimelik gruplar için trigram olarak adlandırılır.

Örneğin, “insan” kelimesi bigram öznitelik çıkarımı uygulandığında [“in”,“ns”,“sa”,“an”] olarak incelenir.

ya da

“yapay zeka dersi” metinsel verisi trigram öznitelik çıkarımı uygulandığında [“yapay zeka dersi”, “zeka dersi yapay”, “dersi yapay zeka”] olarak incelenir.

3.6.4. Character Level yöntemiyle öznitelik çıkarımı

Terim frekansı-ters terim frekansı (TF-IDF) olarak da bilinen bu yöntem metinsel veriler içinde bulunan kelimelerin ne sıklıkla kullanıldığını ve bu sıklık değeri ile önem derecisini bulmak için kullanılan bir öznitelik çıkarım yöntemidir (Korkmaz, 2021).

TF = Karakterin Frekansı / Metinsel Veri Sayısı
IDF = log (Toplam Metin Sayısı / Karakterin Bulunduğu Metin Sayısı)
TF-IDF = TF*IDF olarak hesaplanır.

Metinsel verilerin bulunduğu örnekler Tablo 3.3’te verilmiştir.

Tablo 3.3. Örnek metinsel veriler

Metin 1	elektrik enerjisi üretiminde yenilenebilir kaynaklar önemini her geçen gün arttırmaktadır
Metin 2	yenilenebilir enerji bir ülkenin geleceğidir
Metin 3	sanayileşme de elektrik tutarı önemli bir rol oynamaktadır

Metin 1 içerisinde geçen “yenilenebilir” kelimesinin TF-IDF değerinin hesaplama örneği;

TF = 1/10 = 0,1
IDF = log (3/2) = 0,17
TF-IDF = 0,1*0,17 = 0,017

Tablo 3.3’teki “yenilebilir” karakterinin TF-IDF değeri 0,017 olarak bulunur.

3.7. Sınıflandırma Yöntemleri

Metinsel verilerin ön işleme aşaması ve öznitelik çıkarım süreci tamamlandıktan sonra verilerin hangi sınıfa ait olduğunu bulmak için sınıflandırma çalışması yapılır. Makine öğrenmesinin birçok yöntemiyle, verinin önceden belirtilmiş sınıflardan hangisine ait olduğu bulunarak tahmin ve analiz işlemleri yapılır (Özkan, 2021).

Makine öğrenmesinde kullanılan sınıflandırma yöntemlerinin başlıcaları aşağıda verilmiştir.

3.7.1. Lojistik Regresyon sınıflandırma yöntemi

Regresyon, veriler içerisinde bağımlı değişkenler ile bağımsız değişkenler arasında sayısal bir benzerlik kurarak analiz yapma işlemidir. Lojistik Regresyon, yalnızca regresyon işlemini değil regresyondan sonra sınıflandırma işlemini de yapar. Metinsel verilerde lojistik regresyon kullanılarak duygu analiz yapılırken, olumlu-olumsuz ya da tehdit var-yok olarak sınıflandırma yapılabilir. İhtiyaç halinde "Çok Terimli Regresyon" yapılarak olumlu-nötr-olumsuz diye ikiden fazla sınıflandırma da yapılabilir. Lojistik regresyon yönteminde Formül 3.1'de gösterilen "Sigmoid Fonksiyonu" kullanılarak 0 ile 1 arasında bir değer bularak sınıflandırma işlemi yapılır (Görmez, 2021).

Sigmoid Fonksiyonu = 1 / (1+e^-(Bo + B1x)) (3.1)

Veri bilimcilerin çokça başvurduğu yöntemlerden biridir. Değişken ifadelerle fonksiyon oluşturmak için oldukça fazla kullanılmaktadır (Albayrak, 2020).

3.7.2. Naive Bayes sınıflandırma yöntemi

Makine öğrenmesinde sıklıkla kullanılan bir sınıflandırma yöntemidir. Metinsel veriler içerisinde barınan duyguların tahmininde bulunmak için kullanılır. Örneğin, saldırma amacı bulunan bir blok veya görüşme verilerinden tehdit duygusunun tahmininde kullanılır. Çalışma sırasında karakterlerin frekansını kullanarak sonuç olasılıklarını hesaplar. Naive Bayes sınıflandırma yöntemi formülü, Formül 3.2'de gösterilmiştir (Görmez, 2021).

P(c|x) = (P(x|c)*P(c))/P(x) (3.2)

Naive Bayes yöntemi gözetimli öğrenme uygulamalarında oldukça geniş bir kullanım alanına sahiptir (Öztürk vd., 2020).

3.7.3. Random Forest sınıflandırma yöntemi

Birçok karar ağacının bir modelde oluşumuna izin veren, içeriği ormana benzeyen makine öğrenme sistemine denir (Breiman, 2001). Duygu analizinde kullanılan bir sınıflandırma yöntemi olan Random Forest (Karar Ağacı) sınıflandırma yöntemi, metindeki duygu tahminlerinde karar ağaçları çıkartır ve sonucunda karar ağaçlarını birleştirerek duygu analizi yapar. Random Forest sınıflandırma yöntemi, düğümlerde karar ağaçları gibi önem arz eden noktaları değil, rastsal bir alt küme oluşturarak sınıflandırma yapar. Bu sayede modelin iyileşmesi ve çeşitlilik özelliklerini arttırır (Ballı, 2021).

3.7.4. XGBoost sınıflandırma yöntemi

Aşırı gradyan artırma (XGBoost) sınıflandırma yöntemi, karar ağaçları prensibine dayayan bir sınıflandırma yöntemidir. Bu yöntem aşırı öğrenme (overfiiting) sorununun önüne geçerek sınıflandırma yapmaktadır. Kayıp fonksiyonun, her tahminde gerçekten uzaklaşma farkına bakılarak modeli eğitir. En yüksek tahmini en uygun model olarak kabul eder (Chen ve Guestrin, 2016). Genellikle fazla veriye sahip olmayan verilerin sınıflandırılması için kullanılan denetimli öğrenme yöntemlerinden biridir (Çelik ve Kaplan, 2020).

4. MATERYAL ve YÖNTEM

4.1. Spark NLP ile Oluşturulan Modelde Kullanılan Materyal

Spark NLP ile oluşturulan modelde kullanılan metinsel veriler, Cornell Üniversitesi "Automated Hate Speech Detection and the Problem of Offensive Language" bölümü tarafından hazırlanmıştır (Kaggle, 2023a). Tablo 4.1'de metinsel veri seti dağılımı gösterilmiştir.

Tablo 4.1. Metinsel veri seti dağılımı

Sınıf	Veri Sayısı
Nefret	1430
Saldırı	19190
Tehlike Olmayan	4163
Toplam	**24783**

Tablo 4.1'de 1430 adet nefret, 19190 saldırı ve 4163 adet ise tehlike amacı barındırmayan cümlelerin sayıları model içerisinde ifade edilmiştir. Modelde saldırı amaçlı cümlelerin başarısı hedeflenmiştir. Model eğitimleri Google Colaboratory (Colab, 2023) ortamında gerçekleştirilmiştir.

4.2. LSTM ile Oluşturulan Modelde Kullanılan Materyal

LSTM ile oluşturulan modelde kullanılan ses veri seti, Kaggle internet sitesinden indirilmiştir (Kaggle, 2023b). Toplam 1440 farklı ses dosyası ile model eğitimi geçekleştirilmiştir. Tablo 4.2'de ses veri seti dağılımı gösterilmiştir.

Tablo 4.2. Ses veri seti dağılımı

Duygu	Kadın	Erkek	Toplam
Kızgın	96	96	192
Sakin	96	96	192
İğrenme	96	96	192
Korku	96	96	192
Mutlu	96	96	192
Nötr	48	48	96
Üzgün	96	96	192
Sürpriz	96	96	192
Toplam	**720**	**720**	**1440**

Ses veri seti; kızgın, sakin, iğrenme, korku, mutlu, nötr, üzgün ve sürpriz tonlu 8 farklı duygu barındırmaktadır.

4.3. GRU ile Oluşturulan Modelde Kullanılan Materyal

GRU ile oluşturulan modelde kullanılan görsel veri setleri, Cocodataset internet sitesinden indirilmiştir (Cocodataset, 2023). Şekil 4.1'de görsel veri setinde indeks numarası 1'e karşılık gelen verinin örneklemesi gösterilmiştir. Her bir görüntü için Şekil 4.1'de gösterildiği gibi 5 adet tanımsal cümle bulunmaktadır. Eğitim setinde 118287 adet, test setinde 5000 adet görsel veri bulunmaktadır.

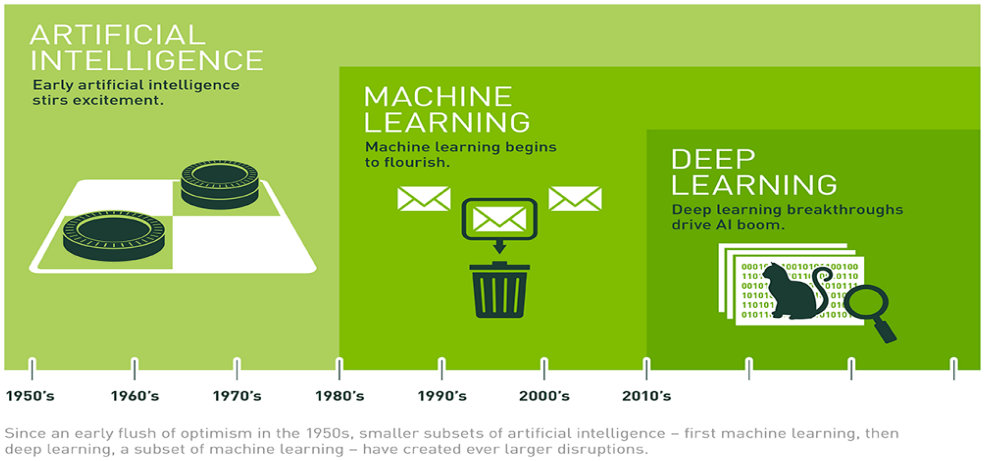
Şekil 4.1. Görsel veri seti örneklemesi

Şekil 4.1'de veri setinden hazır olarak alınan resimde, resim içeriği ve resimle ilgili 5 yorum bulunmaktadır. Zürafanın ağaç yediği bilgisi, resimde zürafanın konum bilgisi, anne ve bebek zürafa olarak aile bilgisi, adet ve çevresel konularda çeşitli bilgileri içermektedir.

4.4. Derin Öğrenme

Günümüz teknoloji dünyasında duygu tanıma, çeviri yapma, karar verme ve sınıflandırma gibi birçok alanda birçok teknolojik kuruluş yapay zeka sistemlerini kullanmaktadırlar. Derin öğrenme, nesne tanıma, konuşma tanıma, doğal dil işleme gibi

alanlarda çok katmanlı yapay sinir ağları kullanan bir yapay zeka yöntemi olup makine öğrenmesinin çeşitlerinden biridir. Şekil 4.2'de yapay zekanın gelişim sürecinde derin öğrenmenin yeri gösterilmiştir. 1950 yıllarında ortaya çıkan yapay zeka, 1980 yılında makine öğrenmesiyle evrilmiş halini almıştır. 2010 yılında ise derin öğrenme modelleri oluşturulmuştur.



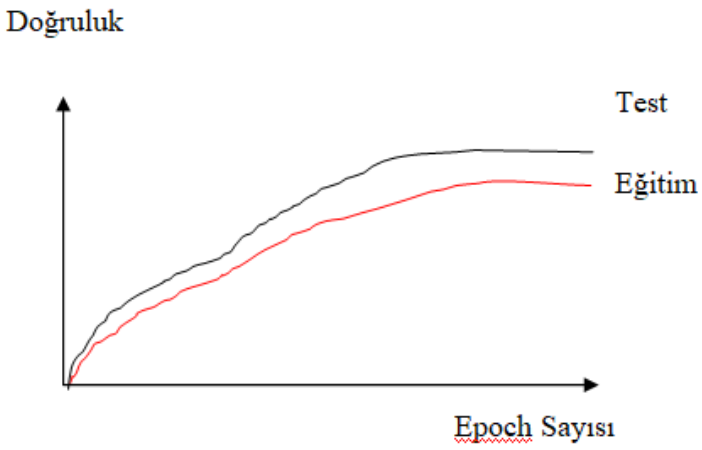
Şekil 4.2. Yapay zekanın gelişim süreci (Nvidia Developer, 2023).

Makine öğrenmesi sisteminde karar verici bir sistem oluştururken, verilerin öznitelik işlemleri kullanıcı tarafından yapılmalıdır. Örneğin, uçak ile gemi resimlerini makine öğrenmesinde ayrıştırmak için bu verilere ait nitelik çıkarmak gerekmektedir. Uçağın kendi ayrışım özellikleri ile geminin ayrışım özellikleri önceden kullanıcı tarafından belirlenmelidir. Derin öğrenme sistemlerinde kullanıcıya ihtiyaç duymadan, şekilleri baz alarak sınıflandırma işlemini model yapabilmektedir (Karaca, 2021).

Derin öğrenmenin mimari yapısı yapay sinir ağlarına çok benzemekte olup en büyük farklılığı gizli katman bölümündeki nöronların fazlalığıdır. Bu sayede derin öğrenme, öznitelik çıkarımı için dışarıdan bir insan kaynağına ihtiyaç duymadan karmaşık problemleri ileri ve geri yayılım özelliğiyle yapay zekanın kendi başına yapmasına olanak sağlamıştır (Ravì vd., 2017). Derin öğrenme yöntemi bilgileri miras alarak öğrenme işlemini gerçekleştirir. Verilen giriş bilgilerini kendi içerisinde bir seçme yapısına tabi tuttuğu için derin öğrenmede diğer yapay zeka yöntemlerine göre daha iyi sonuçlar alınır. Derin sinir ağları ile yüksek başarılı sonuçlar elde edilmesi, son yıllarda derin öğrenmenin popüler olmasına neden olmuştur.

Şekil 4.3. Derin öğrenme modeli

Şekil 4.3'te derin öğrenme modeli verilmiştir. Derin öğrenme modelinin genel yapısı, giriş katmanı, gizli katmanlar ve çıkış katmanından oluşur. Şekilde her bir katmanın sayıları problemin özelliğine göre arttırılıp azaltılabilir. Problemin çözümü için oluşturulan modelin sonuç değerleri ile birlikte grafik yorumlaması yapılarak model evrilmeye devam ettirilebilir. Bu evirme sürecinde modeli oluşturan parametrelerde deneme yanılma yapılması en çok kullanılan yöntemlerdendir. Şekil 4.4'te model eğitim-test grafik örneklendirmesi verilmiştir.

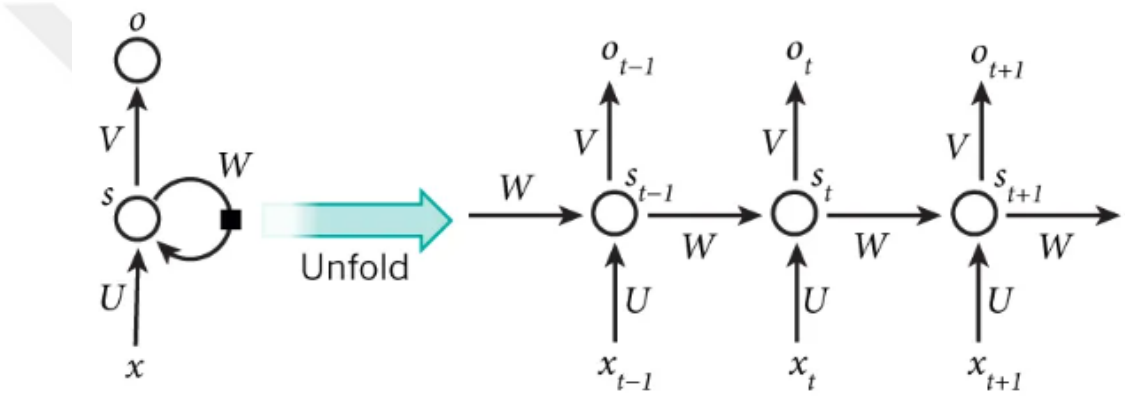


Şekil 4.4. Model eğitim-test grafik örneklendirmesi

Şekil 4.4'te eğitim ve test grafiklerinin birbirine uyumu gösterilmiştir. Modelin doğruluk oranıyla beraber başarısını ispatlayabilmesi için grafik yorumlanmasının yapılabilmesi gerekmektedir. Model bir yerden sonra grafikte birbirinden ayrılıyorsa aşırı uydurma olduğu veya doğruluk oranı yükselmiyorsa az uydurma olduğu söylenebilir. Burada verilerin sayısının ve çeşitliliğinin arttırılması en uygun modelin oluşturulmasında fayda sağlayacaktır. Eğitim ve test eğrilerinin sabitlendiği ya da değişmenin olmadığı yerde sistemin yorulmaması için epoch sayısı belirlenebilir.

4.4.1. RNN yöntemi

Tekrarlayan Sinir Ağı (Recurrent Neural Network, RNN), derin öğrenme biliminin alt dallarından biri olup Elman tarafından tasarlanmıştır. Dil bilimi ile çalışmalar yapanlar için çözüm yolları sunan bir yöntem olmuştur. Cümleleri kategorize etme, duygu tanımlama, sınıflandırma, gerçek zamanlı işlemler gibi birçok alanda kendini ispatlamış bir yöntemdir. Normal sinir ağlarında girdiler birbirinden bağımsız olarak eğitime tabi tutulur. RNN yönteminde ise girdi bilgisinin çıktısı bir önceki çıktı bilgisinden kalıntı alarak çalışmasını devam ettirir (Doğan ve Türkoğlu, 2019). Şekil 4.5'te RNN çalışma modeli verilmiştir.



Şekil 4.5. RNN çalışma modeli (Devhunter, 2023)

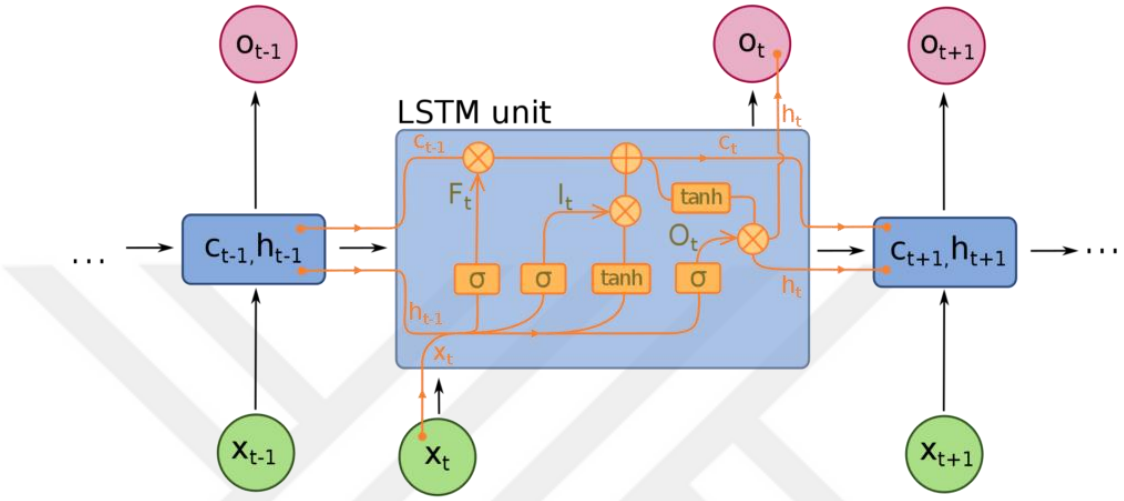
Şekil 4.5'te görüldüğü gibi her katman bir önceki katmandan aldığı bilgiyle eğitimini devam ettirerek modelini oluşturmaktadır.

RNN mimarisinde en ciddi problemlerden biri geri yayılım sırasında türev kullanıldığından, türev sıfıra gittiğinde modelde hata verme sorununun oluşmasıdır. RNN modelinden oluşturulan LSTM ve GRU modelleri bu problemin önüne geçmeyi başarmıştır.

4.4.1.1. LSTM yöntemi

Uzun-Kısa Vadeli Bellek (Long Short-Term Memory, LSTM) sinir ağları olarak literatüre geçen bu yöntem, RNN yönteminde kaybolan gradyan sorununa bir çözüm getirmiştir. Doğal dil işleme biliminin derin öğrenme yöntemlerinden biridir. LSTM ile

hem kısa hem uzun vadeli bilgileri model eğitimi kapsamına alarak, sinir ağlarında bilgi kaybolmasını engellemiştir. Şekil 4.6'da LSTM çalışma modeli gösterilmiştir. Şekilde verilen LSTM çalışma modelinde gösterildiği gibi LSTM modeli girdi bilgilerini işleyen 3 adet hafıza sistemine sahiptir. Dil modeli çözümünde en çok kullanılan yöntemlerden biridir.

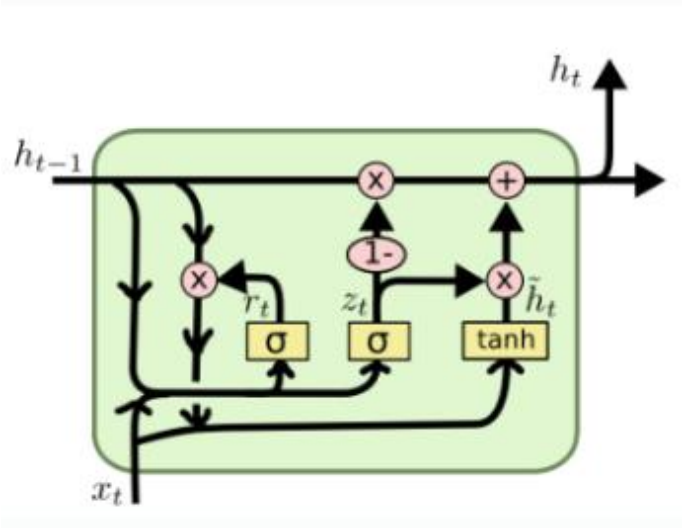


Şekil 4.6. LSTM çalışma modeli (Datakapital, 2023)

Hafıza sisteminde gerekli olan ve olmayan bilgilerin tutulduğu kapılar mevcuttur. Eğer tutulan bilgi önemli ise 1'e önemsiz ise 0'a yaklaştırır. Böylelikle bir önceki hücreden miras alırken bilginin güvenirliğini sürekli kısa-uzun hafıza birimiyle eğitmiş olur (Dayan ve Yılmaz, 2022).

4.4.1.2. GRU yöntemi

Geçitli Tekrarlayan Birim (Gated Recurrent Unit, GRU) olarak bilinen bu yöntem de LSTM yönteminde olduğu gibi RNN sisteminin problemlerini ortadan kaldırmıştır. LSTM yöntemine göre daha hızlı çalışabilen GRU çalışma modeli Şekil 4.7'de gösterilmiştir. Şekilde görüldüğü üzere model, sıfırlama kapısı ve güncelleme kapısı olarak 2 kapıdan oluşmaktadır. LSTM yönteminden daha sonra çıkan bu yöntem daha basit bir mimariye sahiptir. LSTM ile GRU mimarileri denenerek modelde hangisinin başarılı olduğu eğitim sürecine göre değişiklik gösterebilmektedir.



Şekil 4.7. GRU çalışma modeli (Velog, 2023)

GRU çalışma prensibi güncelleme ve sıfırlama kapılarında farklı fonksiyonlar kullanmıştır. GRU yöntemi, LSTM'de olduğu gibi RNN temelli bir mimariye sahiptir (Çetiner, 2022).

4.5. Kelime/Cümle Vektörü Oluşturma Yöntemi

Dijital dünyaya giriş yapan kelimeler bu dünyada kendisini sayılarla ifade eder. İlk önce anlam ifade eden kelimeler, tokenleştirme sürecinden geçerek sayısal bir bilgi haline gelir. Sayısal değere sahip olan kelimelerin anlam ifade edebilmesi, sınıflandırma yapabilmesi, karar verici bir sistemin parçası olabilmesi veya sorulan sorulara cevap verebilmesi için kelime vektörlerine dönüştürülmesi gerekir. Kelime vektörü oluşturma Google Ekibinde çalışan Tomas Mikolov öncülüğüyle geliştirilmiş olup, bu yönteme Word2Vec ismi verilmiştir (Koç vd., 2018).

Bu yöntem iki farklı şekilde ele alınmıştır. Bunlardan ilki Skip-gram yöntemi, diğeri ise CBOW yöntemidir.

Skip-gram yöntemi, cümlenin ortasında bulunan kelimelerden kenarda bulunan kelimeleri tahmin ederek ilişki kurma yöntemine dayanır. Küçük metinlerde CBOW yöntemine göre daha iyi sonuçlar verir. Ayrıca az bulunan kelimeleri de daha iyi temsil eder. Fakat CBOW yöntemine göre yavaştır.

CBOW yöntemi, cümlenin kenarında bulunan kelimeler ile ortada bulunan kelimeleri tahmin edip ilişki kurma yöntemine dayanır. Büyük metin verilerine ihtiyaç duyar ve sık kullanılan kelimeleri temsil etmede iyidir. Skip-gram yöntemine göre daha hızlı eğitim süresi vardır.

İki yöntemi de birer örnekle açıklayacak olursak:

Skip-gram için; "Türkiye'nin en büyük gölü____." cümlesinde altı çizili alana "Van Gölüdür" kelimelerinin gelmesi beklenirken,

CBOW için; "Türkiye'nin ____ gölü Van Gölüdür." cümlesinde altı çizili alana "en büyük" kelimelerinin gelmesi beklenir (Koç vd., 2018).

Kelime vektörlerinin, doğal dil işleme biliminin gelişmesinde büyük payı olmuştur. Dünyada bu alanda çalışan birçok yazılımcı, kelime/cümle vektörü oluşturma yöntemi için metot ve kaynak geliştirmiştir. Bu yöntemlerden en çok kullanılanları aşağıda sunulmuştur.

4.5.1. GloVe ile cümle vektörü oluşturma

Kelime Temsili için Global Vektörler (Global Vectors for Word Representation, GloVe), anlam ifade eden kelime öbeklerini vektör haline çeviren Stanford Üniversitesi tarafından oluşturulup geliştirilen denetimsiz bir öğrenme yöntemidir. Her kelimenin vektör uzunluğu aynı olup, farklı boyutlarda vektör uzunlukları mevcuttur. Kelimeleri vektör haline getirirken kelimelerin anlamsal yakınlıkları ve kosinüs yakınlıklarına göre sınıflandırma yapar (Stanford Üniversitesi, 2023).

Örneğin bir kelimenin doğru sınıfa atama yapabilmesi için; onun yetişkin olması, cinsiyetinin kadın olması ve kraliyet grubundan olması bilgileriyle, onu kraliçe olarak sınıflama yapar. Günümüz teknolojisinde, kelimeler arasında anlam ilişkisi için kullanılan ve tercih edilen yöntemlerden biridir.

4.5.2. BERT ile cümle vektörü oluşturma

BERT (Bidirectional Encoder Representations from Transformers), Google tarafından üretilmiş bir kelime-cümle vektörü oluşturma yöntemidir. Doğal dil işleme biliminin

hem kelimeden cümleye hem de cümleden kelimeye çift yönlü uygulaması olduğundan bu alanda en çok kullanılan yöntemlerden biri denilebilir. Birçok dilde model oluşturabilen vektör sınıflama yöntemidir. Kelime vektör uzunluğu GloVe gibi sabit olmayıp taşıdığı anlama göre uzunluğu değişebilmektedir. En büyük engeli, ön eğitim aşamasında çoklu çalışma yaptığından modeli günlerce eğitebilir. Ön eğitim bitiminde ince ayar aşamasına geçtiğinde saatler içerisinde tamamlayabilir (Github, 2023). Bu sebeple genelde ön eğitimi tamamlanmış hazır verilerde çalışmalar sürdürülmektedir.

4.5.3. USE ile cümle vektörü oluşturma

USE (Universal Sentence Encoder), Google tarafından üretilmiş ön eğitimli bir encoder aracıdır. Kelimeler arasındaki anlam ilişkisinden daha ziyade metinler arasından benzerliğe bakan bir yöntemdir. Doğal dil işleme alanı için yüksek boyutlu vektör uzunluğuna sahiptir (TensorFlow Hub, 2023).

4.5.4. ELMo ile cümle vektörü oluşturma

ELMo (Enough, Let’s Move on), dillerin karmaşık yapısı gereği bir kelime birden fazla anlam ifade edebilmektedir. ELMo, her kelimenin cümle içerisinde farklı anlamlarıyla farklı vektör büyüklüğünde dizin oluşturmaktadır. Derin bir modelleme ağına sahip olan ELMo diğer modellere eklenme imkânı vermektedir. ELMo; duygu analizi, soru ve cevap oluşturma ve metinsel analiz gibi birçok alanda kullanılabilmektedir (Allenai, 2023).

4.5.5. FastText ile cümle vektörü oluşturma

FastText, Facebook ekibi tarafından üretilmiş ve 157 dilde önceden eğitilmiş kelime vektörüne sahiptir. Eğitilmiş modellerinde vektör boyutu 300 olup istenildiğinde küçük boyutlara da indirgeme imkânı vermektedir. Bu modelde, anlam ifade eden simgeler ve diller arasındaki benzerlikler gibi konular ele alınmıştır (FastText, 2023). BERT modelinde olduğu gibi FastText de birçok doğal dil işleme alanında çalışan kişinin tercih ettiği bir cümle vektörü oluşturma modelidir.

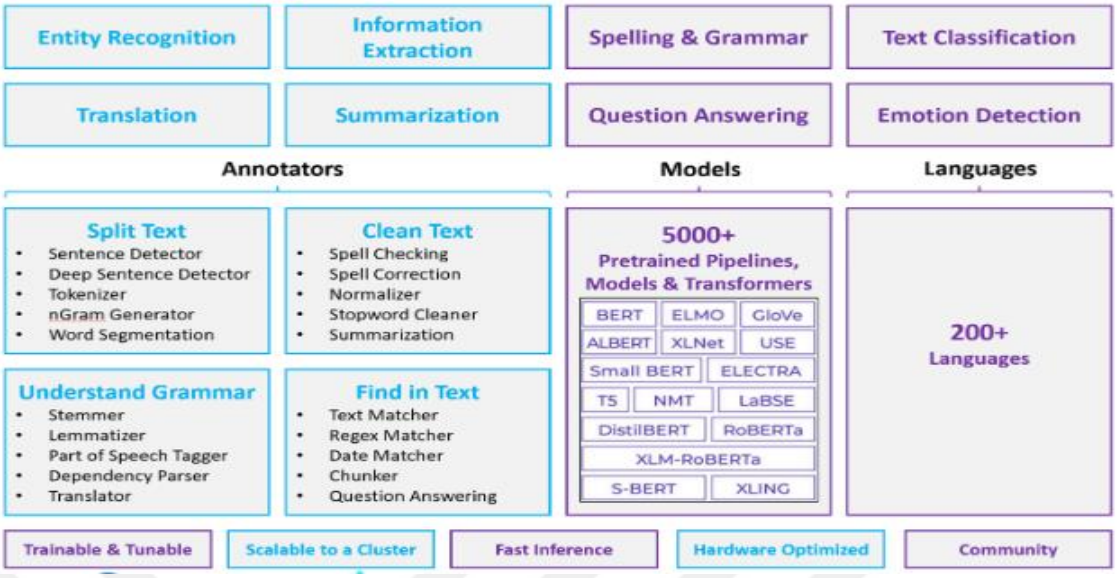
4.6. Spark NLP Yöntemi

Doğal dil işleme bilimi için birçok yöntem ve bu yöntemler kullanılarak birçok kuruluşun çalışmaları bulunmaktadır. Makine öğrenmesi yöntemleri, kelime-cümle vektörü oluşturma yöntemleri, derin öğrenme yöntemleri bunların başlıcalarıdır. Her veri setinde bütün yöntemler denenerek sonuç çıktı skorlarını karşılaştırmak, en iyi yöntemi bulma yollarından biridir. John Snow Labs ve Databricks kuruluşlarının çalışmalarıyla ortaya çıkan Spark NLP (State-of-the-Art Natural Language Processing) ile bütün yöntemleri bir çatı altına almak mümkün olmuştur. Şu an dünyada da en çok kullanılan doğal dil işleme yöntemlerinden biri olarak yerini almıştır (Spark NLP, 2023).

Veri büyüklüğü modelin kendini ispat edip doğrulamasında önemli bir parametredir. Veri setleri büyüdüğünde yüksek bir hesaplama yeteneğine ihtiyaç duyulmaktadır. Spark NLP birçok yöntemi barındırdığından dolayı model sahibine avantaj sağlamakta ve bu nedenle tercih edilmektedir. Nugroho ve arkadaşları yaptıkları çalışmada, BERT yöntemi ile modeli 35 dakikada eğitebilirken, Spark NLP ve BERT yöntemi ile aynı modeli ciddi bir kayıp vermeden 9 dakikada eğitebildiklerini ifade etmişlerdir (Nugroho vd., 2021).

Spark NLP, Pipeline kütüphanesi sayesinde farklı yöntemleri birbiriyle evirmemizi mümkün kılmaktadır. Yüksek hesaplama yöntemine sahip yapay zeka modelleri ortaya çıkartmamızı sağlayan bir yöntemdir.

Şekil 4.8'de Spark NLP uygulama alanları gösterilmiştir. Varlık tanıma, özetleme, sınıflandırma, soru-cevap sistemleri, duygu analizi gibi birçok alanda kullanılabilinmektedir. 200'den fazla dilde, 500'den fazla model oluşturma imkânı sağlamaktadır. Spark NLP yönteminin diğer özellikleri Ek-6'da verilmiştir.

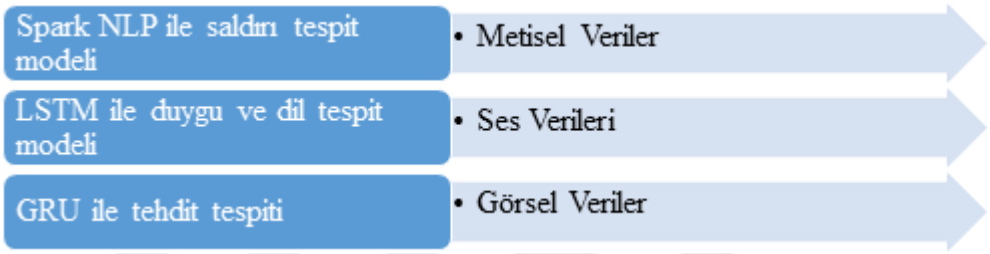


Şekil 4.8. Spark NLP uygulama alanları (Microsoft, 2023)

5. ARAŞTIRMA BULGULARI

5.1. Giriş

Tez çalışmasında doğal dil işleme bilimi ile olası terör tespiti için 3 farklı model oluşturulmuştur. Bu modeller, Spark NLP ile metne çevrilen konuşmalar veya yorumlardan saldırı tespiti, LSTM ile ses tonu özelliği kullanılarak duygu analizi ve dil tespiti ve GRU ile image caption yöntemiyle görüntüleri metne çevirerek olası tehdit tespiti basamaklarından oluşmaktadır. Bu model basamakları Şekil 5.1'de detaylı olarak verilmiştir.



Şekil 5.1. Model basamakları

Şekil 5.1'de her model için doğal dil işleme yöntemleri uygulanmıştır. Spark NLP ve RNN (LSTM, GRU) yöntemleriyle modeller eğitilmiştir. Her model eğitilerek terör ve tehdit unsurlarının tespiti üzerine çalışmalar yapılmıştır. Her modelin iç yapısı, kullanılan veriler ve doğruluk sonuçları bu bölümde ifade edilmiştir.

5.2. Spark NLP ile Metinsel Verilerden Saldırı Tespiti

Eğitim öncesi tokenizer, normalizasyon ve gürültülü verileri temizleme işlemleri tamamlanarak, Spark NLP üzerinden pipeline oluşturulmuştur. Eğitimler dünyada en çok kullanılan kelime/cümle vektörü oluşturma yöntemleri olan ELMo, GloVe ve BERT üzerinden gerçekleştirilmiştir. Spark NLP ile ELMo, GloVe ve BERT üzerinden yapılan eğitimler sonucunda elde edilen model doğruluk oranları ve eğitim süresi değerleri Tablo 5.1'de verilmiştir.

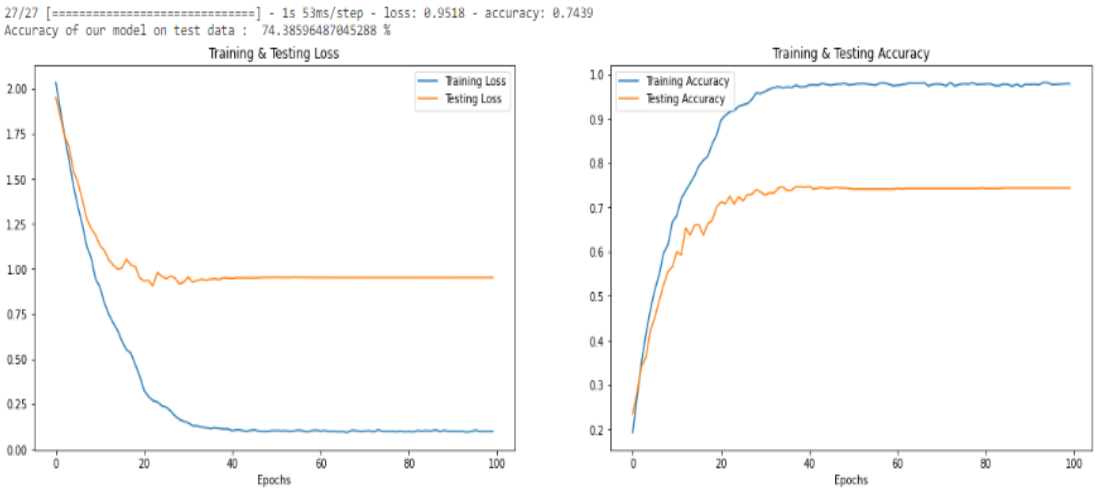
Tablo 5.1. Spark NLP model doğruluk-süre tablosu

Sıra No	Kelime/Cümle Yöntemi	Doğruluk Oranı	Model Eğitim Süresi
1	ELMo	%84	17 dakika
2	GloVe	%85	1 dakika
3	BERT	%84	5 dakika

Spark NLP ile ELMo, GloVe ve BERT üzerinden yapılan eğitimler sonucunda olası saldırı cümlelerinin yakalanmasında en büyük başarıyı GloVe elde etmiştir. GloVe ile diğer yöntemler arasında doğruluk başarı oranı açısından çok fark olmamasına rağmen, GloVe eğitim süresini çok kısa bir sürede gerçekleştirmeyi başarmıştır. Aynı şekilde BERT ile ELMo arasında doğruluk oranlarında fark olmamasına rağmen, eğitim süresi BERT'de ELMo'ya göre çok daha kısa sürede gerçekleşmiştir. Buradaki süreler verilerin sayısına, eğitim yapılan bilgisayarın özelliklerine ve CPU-GPU-TPU yapısına göre değişiklik gösterebilmektedir. Normalde bu yöntemler Spark NLP kullanılmadan eğitilmeye başlandığında, bu sürelerin çok daha uzun olduğu görülmüştür. Analiz sırasında kullanılan pipeline model hazırlıkları ve eğitim kodları Ek-1'de gösterilmiştir.

5.3. LSTM ile Ses Verilerinden Duygu ve Dil Tespiti

Yapılan araştırmalar niyet analizinde ses tonunun kelimelerden daha önemli bir payının olduğunu göstermektedir. Sinirli, korkulu ya da iğrenme temalı bir sesin saldırı amacı taşımasının daha yüksek ihtimal olduğu düşünülmektedir. Oluşturulan modelde hem ses tonu analizi hem de dil tespiti yapılarak olası terör-tehdit niyetini bulma amacı edinilmiştir. Modelde doğal dil işlemenin derin öğrenme alanında LSTM yöntemi kullanılmıştır. Model oluşturma aşamasında parametreler her eğitim sonucunda değiştirilerek en yüksek performanslı sonuç olan %74 doğruluk oranına ulaşılmıştır. Şekil 5.2'de ses analizi eğitim ve test grafiği verilmiştir.



Şekil 5.2. Ses analizi eğitim ve test grafiği

100 epoch ile gerçekleştirilen model eğitimleri sonucunda oluşan kayıp ve kazanç grafiğinde Şekil 5.2'de görüldüğü gibi stabilizasyon elde edilmiştir. Şekilde de gösterildiği gibi yapılan eğitimlerde 40 epoch sonucunda da yaklaşık aynı doğruluk oranına erişilmiştir.

Tablo 5.2. Ses tahmin sonuçları

Tahmin Sonucu	Gerçek Ses Etiketi
surprise	surprise
disgust	disgust
angry	angry
angry	angry
surprise	surprise
happy	happy
fear	fear
happy	happy
happy	neutral
disgust	disgust

Tablo 5.2'de modelin doğruluğunu örnekleyebilmek için yapılan çalışmada sadece mutlu ses tonunu nötr hatası dışında diğer bütün tahminlerde doğru sonuca ulaşılmıştır. Ayrıca olası tehdit ve terör bölgesinde haberleşme ağında kullanılan dilin de tespiti doğal dil işleme bilimi ile yapılarak tespit süresinin kısalacağı düşünülmektedir. Sesi metne çeviren program kodları Ek-4'te ve dil tespiti yapan program kodları Ek-5'te gösterilmiştir. Ses analizi için kullanılan modelin kodları ise Ek-2'de gösterilmiştir.

5.4. GRU ile Görsel Verilerden Tehdit Tespiti

Doğal dil işleme duygu analizinin sadece metinsel veriler üzerinden değil görüntüler üzerinden de yapılmasına olanak sağlamıştır. Böylelikle görsel verilerin içeriklerini metinsel verilere dönüştürerek olası terör-tehdit unsurlarının tespitinin yapılmasına olanak sağlar.

GRU ile yapılan model eğitimleri, Google Colaboratory ortamında gerçekleştirilmiş olup, toplam 3 saat 17 dakika sürmüştür. Parametreler üzerinde değişiklikler yapılarak %71 doğruluk oranına ulaşılmıştır. Şekil 5.3'te terör temalı örnek bir resim gösterilmiştir.

Şekil 5.3. Terör temalı örnek resim

İnternet üzerinden alınan Şekil 5.3'te gösterilen resim, model eğitimi sonunda tahmin edilmeye çalışılmıştır. Tahmin sonucu "three men in uniform with guns in their hands", "ellerinde silahları olan üniformalı üç adam" tahminine başarılı bir şekilde ulaşılmıştır. Normalde resimde 5 kişi olmasına rağmen program 3 kişiyi algılamış olup onların bilgilerini metne çevirmeyi başarmıştır.

Şekil 5.4'te tehdit temalı örnek bir resim gösterilmiştir. Şekil 5.4'te gösterilen resim Ekinlaw internet sitesinden alınmış olup (Ekinlaw, 2023), tehdit figürleri barındırmaktadır. Modelin resmi tahmin etmesi sonucunda "a man holding a knife in his right hand", "sağ elinde bıçak tutan bir adam" tahminiyle model yine başarılı bir sonuç ortaya koymuştur.

Şekil 5.4. Tehdit temalı örnek resim

Şekil 5.4'te insan suretleri puslandırılmasına karşılık yine de insan figürünü tespit etmiş olup elindeki tehdit unsurunu da başarıyla ifade etmiştir. Dikkat edildiğinde, arkada başka insanların olduğu da görülmektedir. Arkadaki kişilerin puslandırma seviyesi fazla olduğundan, model bu kişileri tespit etmekte başarılı olamamıştır. Buda gösteriyor ki, modelin başarısı için encode edilip dijital dünyaya aktarılan verilerle sağlıklı bir eğitim aşaması gerçekleştirildikten sonra, model çıktısı verilirken temiz görüntülere ihtiyaç duyulmaktadır.

Şekil 5.5'te öfke temalı örnek bir resim gösterilmiştir. Şekilde gösterilen resim Dentiss internet sitesinden alınmış olup (Dentiss, 2023), öfke duygusuna sahip bir insanın figürünü barındırmaktadır. Modelin tahmini "a man is talking on a cellphone", "telefonda konuşan bir adam" yorum tahminiyle sonuçlanmıştır. Tahmin sonucunda öfke ile ilgili bir bilgi edinilmemiştir. Cocodataset içerisinde öfke figürlerine ait resim ve yorumlar bulunmadığından tahmin sonucunda başarısızlık meydana gelmiştir. Dışarıdan resim ve yorumlar eklenerek yapılacak bir eğitimde başarılı sonuçlar elde edilebildiği diğer resimlerden anlaşılmaktadır. Tez çalışmasında terör-tehdit unsurlarının tespit edilmesi metinsel, işitsel ve görüntüsel olarak ele alındığı için, görüntüsel olarak başarıya ulaşamayan bu verinin konuşma metinleri ve ses tonu üzerinden bizi başarıya götüreceği yukarıda yapılan çalışmalarda ispat edilmiştir.

Şekil 5.5. Öfke temalı örnek resim

Görüntüden metin çıkaran programın tahminini gerçekleştiren fonksiyon kodları Ek-3'te gösterilmiştir.

6. SONUÇ ve ÖNERİLER

Bu tez çalışmasında, doğal dil işleme yöntemleri ile terör tehdit unsurlarının tespiti için hem metin hem ses hem de görüntü verileri üzerinde çalışan modeller oluşturulmuştur. Model eğitimlerinde doğruluk oranları gösterilmiş olup, yöntem karşılaştırmaları yapılmıştır.

İlk modelde metinsel veriler üzerinde çalışma yapılmıştır. Spark NLP Kütüphanesi ile oluşturulan bu modelde 3 farklı kelime/cümle vektörü oluşturma yöntemi kullanılmıştır. Bu yöntemler doğal dil işleme biliminde en yaygın kullanılan ELMo, GloVe ve BERT yöntemleridir. Bu yöntemler arasında veri setinde hem doğruluk oranı hem de süre olarak en başarılı yöntemin GloVe olduğu eğitim sonucundaki çıktılardan elde edilmiştir. GloVe kelime/cümle vektör oluşturma yöntemi, Google Colaboratory ara yüzü üzerinden 1 dakika içerisinde modeli oluşturmayı başararak %85 doğruluk oranına ulaşmıştır.

İkinci modelde ses verileri üzerinde çalışma yapılmıştır. 1140 farklı ses dosyasına ait 8 farklı duygu incelenmiştir. RNN yönteminin geliştirilmesi ile elde edilen LSTM yöntemi ile model eğitimi gerçekleştirilmiştir. Kaggle internet sitesinin code menüsü ara yüzü ile model kod blokları yazılmış olup doğruluk başarısı için parametreler değiştirilerek eğitim birçok kez gerçekleştirilmiştir. Gerçekleştirilen eğitimler arasında en başarılı sonuç %74 doğruluk oranı ile elde edilmiştir.

Üçüncü modelde görsel veriler ve bu görsellerin yorumları üzerinde çalışma yapılmıştır. Modelde yine bir RNN yöntemi olan GRU yöntemi kullanılmıştır. Modelde image caption olarak isimlendirilen görüntüleri metne çevirme yöntemi uygulanmıştır. Google Colaboratory ara yüzü üzerinden model kurulumu, eğitimi ve tahmini gerçekleştirilmiş olup %71'lik doğruluk oranı elde edilmiştir.

Doğal dil işleme bilimi kullanılarak oluşturulan modellerden elde edilen doğruluk oranları incelendiğinde, en başarılı sonuçların metinsel verilerden elde edildiği görülmektedir. Bunun sebebi metinsel verilerin yapısının ses ve görüntü verilerinin yapısına göre daha kararlı özelliğe sahip olmasıdır. Doğal dil işleme biliminin metinsel verilerde daha başarılı olduğu yapılan çalışmalarda da bilinmektedir. Ses ve görüntü

verileri üzerinden oluşturulan modeller içerisinde, tez çalışmasındaki modellerin doğruluk oranları ortalamanın üstünde bir değere sahiptir.

Bu tez çalışması gösteriyor ki, doğal dil işleme ile olası terör bölgelerinde aynı anda bütün frekanslar insansız bir şekilde dinlenip metinsel verilere dönüştürüldüğünde, terör-tehdit unsurlarının tespitinin yapılabilmesi mümkün olacaktır. Aynı zamanda terör-tehdit unsurlarının olası kullandıkları dil tespiti de yapılabilmektedir. Ayrıca görsel verilerin de metinsel dönüşümü sağlanarak, terör-tehdit eylemlerinin gerçekleşmeden sonlanabileceğini göstermek tez çalışmasının hedeflerindendir.

Metinsel verilerin incelenmesinde Kaggle internet sitesindeki çalışmalar ortalama %81 oranında başarıya sahipken bu çalışmada %85 oranında bir başarıya, ses verilerinin incelenmesinde Kaggle internet sitesindeki çalışmalar ortalama %72 oranında başarıya sahipken bu çalışmada %74 oranında bir başarıya ve görsel verilerin metne çevirme basamağında Kaggle internet sitesindeki çalışmalar ortalama %65 oranında başarıya sahipken bu çalışmada %71 oranında bir başarıya sahip olunmuştur.

Tez çalışmasında gelişimine devam eden doğal dil işlemi yöntemleri kullanılarak terör tehdit unsurlarının tespiti ele alınmıştır. Doğal dil işleme alanında metinsel verilerin incelenmesi, ses verilerinin incelenmesi ve görsel verilerin metinsel verilere dönüşümü konusunda literatüre katkı sağlanması amaçlanmıştır. Gelecekte farklı yöntemler ve farklı veri setleri kullanılarak çalışmalar yapılması düşünülmektedir.

KAYNAKLAR

Adalı, E. (2012) "Doğal dil işleme", ***Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi***, 5(2).

Aksoy, N. (2021) "Türkçe dilinde yapılmış açık uçlu sınavların doğal dil işleme ile otomatik olarak değerlendirilmesi", Yüksek Lisans Tezi, ***Balıkesir Üniversitesi, Fen Bilimleri Enstitüsü, Elektrik-Elektronik Mühendisliği Ana Bilim Dalı***, Balıkesir.

Albayrak, A. (2020) "Doğal dil işleme teknikleri kullanılarak disiplinler arası lisansüstü ders içeriği hazırlanması", ***Bilişim Teknolojileri Dergisi***, 13(4), 373-383.

Allenai (2023) "ELMo", https://allenai.org/allennlp/software/elmo Erişim Tarihi: 04.05.2023.

Atlı, Y. ve İlhan, N. (2021) "Duygu Analizi İçin Yeni Bir Sözlük; NAYALex Duygu Sözlüğü", ***Avrupa Bilim ve Teknoloji Dergisi***, 27, 1050-1060.

Ballı, Ç. (2021) "Doğal dil işleme ile Türkçe içerikli paylaşımlardan sosyal medya kullanıcılarının duygu analizi", Yüksek Lisans Tezi, ***Ankara Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Ana Bilim Dalı***, Ankara.

Bostancı, B. ve Albayrak, A. (2021) "Duygu analizi ile kişiye özel içerik önermek", ***Veri Bilimi***, 4(1), 53-60.

Breiman, L. (2001) "Random forests", ***Machine Learning***, 45, 5-32.

Canım, M. (2019) "Eski Dilde Kullanılan Sözcükler Arasındaki Anlamsal Yakınlıkların Doğal Dil İşleme Yöntemleriyle Tespiti", ***Gümüşhane Üniversitesi Fen Bilimleri Dergisi***, 9(3), 536-546.

Chen, T. and Guestrin, C. (2016, August) "Xgboost: A scalable tree boosting system", ***In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining***, (pp. 785-794).

Cocodataset (2023) "COCO dataset", https://cocodataset.org/#download/ Erişim Tarihi: 25.05.2023.

Colab (2023) "Google Colaboratory", https://colab.research.google.com Erişim Tarihi: 20.04.2023.

Çelik, Ö. ve Kaplan, G. (2020) "Yeniden örnekleme teknikleri kullanarak sms verisi üzerinde metin sınıflandırma çalışması", ***Erciyes Üniversitesi Fen Bilimleri Enstitüsü Fen Bilimleri Dergisi***, 36(3), 433-442.

Çetiner, H. (2022) "Multi-Label Text Analysis with GRU and CNN Based Hybrid Deep Learning Model", ***EasyChair***, 8449.

Datakapital (2023) “Yinelenen Sinir Ağları Nedir?”, https://datakapital.com/blog/yinelenen-sinir-aglari-nedir/ Erişim Tarihi: 02.07.2023.

Dayan, A. ve Yılmaz, A. (2022) “Doğal dil işleme ve derin öğrenme algoritmaları ile makine dili modellemesi”, ***Dicle Üniversitesi Mühendislik Fakültesi Mühendislik Dergisi***, 13(3), 467-475.

Delibas, A. (2008) “Doğal dil işleme ile Türkçe yazım hatalarının denetlenmesi”, Yüksek Lisans Tezi, ***İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Ana Bilim Dalı***, İstanbul.

Demir, O. (2012) “Yapay zekâ”, ***Dokuz Eylül Üniversitesi Edebiyat Fakültesi Dilbilim Bölümü***, 3.

Deng, L. and Yu, D. (2014) “Deep learning: methods and applications”, ***Foundations and Trends in Signal Processing***, 7(3–4), 197-387.

Dentiss (2023) “Doktora Tehdit Cezasız Kalmadı”, https://www.dentiss.com/doktora-tehdit-cezasiz-kalmadi-y2924.html Erişim Tarihi: 06.07.2023.

Devhunter (2023) “Tekrarlayan Sinir Ağları Eğitimi, Bölüm 1 – RNN’lere Giriş”, https://devhunteryz.wordpress.com/2018/07/09/tekrarlayan-sinir-aglari-egitimi-bolum-1-rnnlere-giris/ Erişim Tarihi: 02.07.2023.

Doğan, F. ve Türkoğlu, İ. (2019) “Derin öğrenme modelleri ve uygulama alanlarına ilişkin bir derleme”, ***Dicle Üniversitesi Mühendislik Fakültesi Mühendislik Dergisi***, 10(2), 409-445.

Doğuç, Ö., Aytaç, Ö. B. ve Silahtaroğlu, G. (2020) “Lemmatizer: Akıllı Türkçe kök bulma yöntemi”, ***Turkish Studies-Information Technologies and Applied Sciences***.

Ekinlaw (2023) “Tehdit Suçu Nedir? Şartları Ve Cezası Neler?”, https://www.ekinlaw.com/tehdit-sucu-nedir-sartlari-ve-cezasi-neler/ Erişim Tarihi: 06.06.2023.

Ekim, H. E. ve İnner, A. B. (2021) “Duygu analizi ve fikir madenciliği uygulamaları üzerine literatür taraması”, ***Kahramanmaraş Sütçü İmam Üniversitesi Mühendislik Bilimleri Dergisi***, 24(2), 93-114.

FastText (2023) “Word vectors for 157 languages”, https://fasttext.cc/docs/en/crawl-vectors.html Erişim Tarihi: 06.05.2023.

Github (2023) “What is BERT?”, https://github.com/google-research/bert#what-is-bert Erişim Tarihi: 04.05.2023.

Görmez, B. (2021) “Adli bilişimde makine öğrenmesi: makine öğrenmesi algoritmaları ile terör olaylarının tahmin edilmesi çalışması”, Yüksek Lisans Tezi, ***Ankara***

Üniversitesi, Sağlık Bilimleri Enstitüsü, Disiplinlerarası Adli Bilimler Ana Bilim Dalı, Ankara.

Hark, C., Myo, A., Seyyarer, A., Myo, G., Uçkan, T., Myo, B. and Karci, A. (2017, September) "Doğal dil İşleme yaklaşimlari ile yapisal olmayan dökümanlarin benzerliği", ***In 2017 International Artificial Intelligence and Data Processing Symposium (IDAP)***, (pp. 1-6). IEEE.

İlhan, N. ve Sağaltıcı, D. (2020) "Twitter'da duygu analizi", ***Harran Üniversitesi Mühendislik Dergisi***, 5(2), 146-156.

Kaggle (2023a) "Hate Speech and Offensive Language Dataset", https://www.kaggle.com/datasets/mrmorj/hate-speech-and-offensive-language-dataset/ Erişim Tarihi: 10.05.2023.

Kaggle (2023b) "Speech Emotion Recognition using LSTM", https://www.kaggle.com/code/blitzapurv/speech-emotion-recognition-using-lstm/input/ Erişim Tarihi: 20.05.2023.

Karaca, A. (2021) "Derin öğrenme ile Türkçe haber metinlerine başlık üretme", Yüksek Lisans Tezi, ***Trakya Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Ana Bilim Dalı***, Edirne.

Khalid, S., Khalil, T. and Nasreen, S. (2014, August) "A survey of feature selection and feature extraction techniques in machine learning", ***In 2014 science and information conference***, (pp. 372-378). IEEE.

Khyani, D., Siddhartha, B. S., Niveditha, N. M. and Divya, B. M. (2020) "An interpretation of lemmatization and stemming in natural language processing", ***Journal of University of Shanghai for Science and Technology***, 22(10), 350-357.

Koç, E., Çalışkan, S., Yazıcıoğlu, S. A., Demirci, U. ve Kuş, Z. (2018) "Yapay Sinir Ağları, Kelime Vektörleri ve Derin Öğrenme Uygulamaları".

Kontuk, R. ve Turan, M. (2020) "NLP Kullanılarak Haberlerin Yaş Gruplarına Göre Sınıflandırılması", ***Gazi University Journal of Science Part C: Design and Technology***, 8(2), 372-382.

Korkmaz, S. (2021) "Makine öğrenme yöntemleri kullanılarak doğal dil işleme tabanlı şair tanıma", Yüksek Lisans Tezi, ***Erciyes Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Ana Bilim Dalı***, Kayseri.

Küçük, D. ve Arıcı, N. (2018) "Doğal dil işlemede derin öğrenme uygulamaları üzerine bir literatür çalışması", ***Uluslararası Yönetim Bilişim Sistemleri ve Bilgisayar Bilimleri Dergisi***, 2(2), 76-86.

Microsoft (2023) "Apache Spark as a customized NLP framework", https://learn.microsoft.com/en-us/azure/architecture/data-guide/technology-choices/natural-language-processing Erişim Tarihi: 02.07.2023.

Nugroho, K. S., Sukmadewa, A. Y. and Yudistira, N. (2021, September) "Large-scale news classification using bert language model: Spark nlp approach", ***In Proceedings of the 6th International Conference on Sustainable Information Engineering and Technology***, (pp. 240-246).

Nvidia Developer (2023) "Deep Learning", developer.nvidia.com/deep-learning/ Erişim Tarihi: 02.07.2023.

Oflazer, K. (2016) "Türkçe ve doğal dil işleme", ***Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi***, 5(2).

Oğuzlar, A. (2011) "Temel metin madenciliği", ***Dora***.

Özkan, B. (2021) "Dialog intent classification using NLP methods", Yüksek Lisans Tezi, ***Bahçeşehir Üniversitesi, Lisansüstü Eğitim Enstitüsü, Bilgisayar Mühendisliği Ana Bilim Dalı***, İstanbul.

Özmutlu, A. E. (2021) "Doğal dil işleme", ***Bilgisayar Bilimlerinde Teorik Ve Uygulamalı Araştırmalar***, 129.

Öztürk, A., Durak, Ü. ve Badıllı, F. (2020) "Twitter verilerinden doğal dil işleme ve makine öğrenmesi ile hastalik tespiti", ***Konya Journal of Engineering Sciences***, 8(4), 839-852.

Polat, H. ve Yılmaz, A. (2022) "Tripadvisor kullanıcılarının Türkçe ve İngilizce yorumları kapsamında duygu analizi yöntemlerinin karşılaştırmalı analizi", ***Abant Sosyal Bilimler Dergisi***, 22(2), 901-916.

Python (2022) "What are Word Vectors?", https://pypi.org/project/word-vectors/#what-are-word-vectors Erişim Tarihi: 24.05.2022.

Ravì, D., Wong, C., Deligianni, F., Berthelot, M., Andreu-Perez, J., Lo, B. and Yang, G. Z. (2017) "Deep learning for health informatics", ***IEEE journal of biomedical and health informatics***, 21(1), 4-21.

Seker, S. E. (2016) "Duygu Analizi (Sentimental Analysis)", ***YBS Ansiklopedi***, 3(3), 21-36.

Spark NLP (2023) "Spark NLP State of the Art Natural Language Processing", https://sparknlp.org Erişim Tarihi: 17.05.2023.

Stanford Üniversitesi (2023) "GloVe: Global Vectorsfor Word Representation", https://nlp.stanford.edu/projects/glove/ Erişim Tarihi: 04.05.2023.

Taşkıran, S. F. (2021) "Doğal dil işleme ile akademik metinlerin kümelenmesi", Yüksek Lisans Tezi, ***Konya Teknik Üniversitesi, Lisansüstü Eğitim Enstitüsü, Bilgisayar Mühendisliği Ana Bilim Dalı***, Konya.

TensorFlow Hub (2023) "universal-sentence-encoder", https://tfhub.dev/google/collections/universal-sentence-encoder/1 Erişim Tarihi: 04.05.2023.

Toğaçar, M., Eşidir, K. A. ve Ergen, B. (2021) "Yapay Zekâ Tabanlı Doğal Dil İşleme Yaklaşımını Kullanarak İnternet Ortamında Yayınlanmış Sahte Haberlerin Tespiti", ***Journal of Intelligent Systems: Theory and Applications***, 5(1), 1-8.

Toprak, A. (2019) "Doğal dil işleme ile İngilizce otomatik sözlük oluşturma", Yüksek Lisans Tezi, ***İstanbul Ticaret Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Ana Bilim Dalı***, İstanbul.

Tuzcu, S. (2020) "Çevrimiçi Kullanıcı Yorumlarının Duygu Analizi ile Sınıflandırılması", ***Eskişehir Türk Dünyası Uygulama ve Araştırma Merkezi Bilişim Dergisi***, 1(2), 1-5.

Türkiye Cumhuriyeti Cumhurbaşkanlığı (2022) "Dijital Dönüşüm Ofisi", https://cbddo.gov.tr/sss/yapay-zeka/ Erişim Tarihi: 25.05.2022.

Wang, D., Su, J. and Yu, H. (2020) "Feature extraction and analysis of natural language processing for deep learning English language", ***IEEE Access***, 8, 46335-46345.

Velog (2023) "GRU with pytorch", https://velog.io/tags/GRU/ Erişim Tarihi: 02.07.2023.

Yılmaz, H. ve Yumuşak, S. (2021) "Açık kaynak doğal dil işleme kütüphaneleri", ***İstanbul Sabahattin Zaim Üniversitesi Fen Bilimleri Enstitüsü Dergisi***, 3(1), 81-85.

Yılmaz, M. C. ve Orman, Z. (2021) "LSTM Derin Öğrenme Yaklaşımı ile Covid-19 Pandemi Sürecinde Twitter Verilerinden Duygu Analizi", ***Acta Infologica***, 5(2), 359-372.

Yilmaz, Ş. Ş., Özer, İ. ve Gökçen, H. (2021, February) "Türkçe metinlerde derin öğrenme yöntemleri kullanılarak duygu analizi", ***In International Symposium of Scientific Research and Innovative Studies***, (vol. 22, p. 25).

Yurt, E. A. (2015) "Türkçe metinlerde duygu analizi", Yüksek Lisans Tezi, ***Maltepe Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Ana Bilim Dalı***, İstanbul.

EKLER

Ek-1. Spark NLP Kod Blokları

#Pipeline hazırlanması

```
document_assembler = DocumentAssembler() \
.setInputCol("tweet") \
.setOutputCol("document") \
.setCleanupMode("shrink")

tokenizer = Tokenizer() \
.setInputCols(["document"]) \
.setOutputCol("token") \
.setSplitChars(['-']) \
.setContextChars(['(', ')', '?', '!', '#', '@'])

normalizer = Normalizer() \
.setInputCols(["token"]) \
.setOutputCol("normalized")\
.setCleanupPatterns(["[^\w\d\s]"])

stopwords_cleaner = StopWordsCleaner()\
.setInputCols("normalized")\
.setOutputCol("cleanTokens")\
.setCaseSensitive(True)

lemma = LemmatizerModel.pretrained('lemma_antbnc') \
.setInputCols(["cleanTokens"]) \
.setOutputCol("lemma")

preproc_pipeline = Pipeline(
  stages=[document_assembler,
          tokenizer,
          normalizer,
          stopwords_cleaner,
          lemma])
```

#Glove için pipeline hazırlanması

```
document_assembler = DocumentAssembler() \
.setInputCol("tweet") \
.setOutputCol("document") \
.setCleanupMode("shrink")

tokenizer = Tokenizer() \
.setInputCols(["document"]) \
.setOutputCol("token") \
.setSplitChars(['-']) \
.setContextChars(['(', ')', '?', '!', '#', '@'])

normalizer = Normalizer() \
.setInputCols(["token"]) \
.setOutputCol("normalized")\
.setCleanupPatterns(["[^\w\d\s]"])

stopwords_cleaner = StopWordsCleaner()\
.setInputCols("normalized")\
.setOutputCol("cleanTokens")\
.setCaseSensitive(True)

lemma = LemmatizerModel.pretrained('lemma_antbnc') \
.setInputCols(["cleanTokens"]) \
.setOutputCol("lemma")

glove_embeddings = WordEmbeddingsModel().pretrained() \
.setInputCols(["document",'lemma'])\
.setOutputCol("embeddings")\
.setCaseSensitive(True)

embeddingsSentence = SentenceEmbeddings() \
.setInputCols(["document", "embeddings"]) \
.setOutputCol("sentence_embeddings") \
.setPoolingStrategy("AVERAGE")

classsifierdl = ClassifierDLApproach()\
.setInputCols(["sentence_embeddings"])\
.setOutputCol("classing")\
.setLabelColumn("class")\
.setMaxEpochs(10)\
.setLr(0.001)\
.setBatchSize(8)\
.setEnableOutputLogs(True)


Glove_clf_pipeline = Pipeline(
    stages=[document_assembler,
            tokenizer,
            normalizer,
```

```
        stopwords_cleaner,
        lemma,
        glove_embeddings,
        embeddingsSentence,
        classsifierdl])
```

#Glove için modelin eğitilmesi

```
Glove_pipelineModel = Glove_pipeline.fit(train_df)
```

#Elmo için pipeline hazırlanması

```
document_assembler = DocumentAssembler() \
.setInputCol("tweet") \
.setOutputCol("document") \
.setCleanupMode("shrink")

tokenizer = Tokenizer() \
.setInputCols(["document"]) \
.setOutputCol("token") \
.setSplitChars(['-']) \
.setContextChars(['(', ')', '?', '!', '#', '@'])

normalizer = Normalizer() \
.setInputCols(["token"]) \
.setOutputCol("normalized")\
.setCleanupPatterns(["[^\w\d\s]"])

stopwords_cleaner = StopWordsCleaner()\
.setInputCols("normalized")\
.setOutputCol("cleanTokens")\
.setCaseSensitive(True)

lemma = LemmatizerModel.pretrained('lemma_antbnc') \
.setInputCols(["cleanTokens"]) \
.setOutputCol("lemma")

elmo_embeddings = ElmoEmbeddings.pretrained('elmo')\
.setInputCols(["document", "token"])\
.setOutputCol("embeddings")\
.setPoolingLayer('elmo')

embeddingsSentence = SentenceEmbeddings() \
.setInputCols(["document", "embeddings"]) \
.setOutputCol("sentence_embeddings") \
.setPoolingStrategy("AVERAGE")

classsifierdl = ClassifierDLApproach()\
.setInputCols(["sentence_embeddings"])\
.setOutputCol("classing")\
.setLabelColumn("class")\
.setMaxEpochs(10)\
.setLr(0.001)\
.setBatchSize(8)\
.setEnableOutputLogs(False)
#.setOutputLogsPath('logs')

elmo_pipeline = Pipeline(
   stages=[document_assembler,
           tokenizer,
           normalizer,
```

```
        stopwords_cleaner,
        lemma,
        elmo_embeddings,
        embeddingsSentence,
        classsifierdl])
```

#Elmo için modelin eğitilmesi

```
Elmo_pipelineModel = elmo_pipeline.fit(train_df)
```

#Bert için pipeline hazırlanması

```
document_assembler = DocumentAssembler() \
.setInputCol("tweet") \
.setOutputCol("document") \
.setCleanupMode("shrink")

tokenizer = Tokenizer() \
.setInputCols(["document"]) \
.setOutputCol("token") \
.setSplitChars(['-']) \
.setContextChars(['(', ')', '?', '!', '#', '@'])

normalizer = Normalizer() \
.setInputCols(["token"]) \
.setOutputCol("normalized")\
.setCleanupPatterns(["[^\w\d\s]"])

stopwords_cleaner = StopWordsCleaner()\
.setInputCols("normalized")\
.setOutputCol("cleanTokens")\
.setCaseSensitive(False)

lemma = LemmatizerModel.pretrained('lemma_antbnc') \
.setInputCols(["cleanTokens"]) \
.setOutputCol("lemma")

bert_embeddings = BertEmbeddings().pretrained(name='bert_base_cased', lang='en') \
.setInputCols(["document",'token'])\
.setOutputCol("embeddings")

embeddingsSentence = SentenceEmbeddings() \
.setInputCols(["document", "embeddings"]) \
.setOutputCol("sentence_embeddings") \
.setPoolingStrategy("AVERAGE")

classsifierdl = ClassifierDLApproach()\
.setInputCols(["sentence_embeddings"])\
.setOutputCol("classing")\
.setLabelColumn("class")\
.setMaxEpochs(10)\
.setLr(0.001)\
.setBatchSize(8) \
.setEnableOutputLogs(True)
#.setOutputLogsPath('logs')

bert_pipeline = Pipeline(
   stages=[document_assembler,
           tokenizer,
           normalizer,
           stopwords_cleaner,
```

```
        lemma,
        bert_embeddings,
        embeddingsSentence,
        classsifierdl])
```

#Bert için modelin eğitilmesi

```
Bert_pipelineModel = bert_pipeline.fit(train_df)
```

Ek-2. LSTM Kod Blokları

#Ses analizi için model oluşturulması

```
model=Sequential()
model.add(TimeDistributed(Conv1D(16, 3, padding='same', activation='tanh'),
input_shape=input_shape))
model.add(TimeDistributed(BatchNormalization()))
#model.add(TimeDistributed(MaxPooling2D((2,1))))
model.add(TimeDistributed(Flatten()))
model.add(LSTM(64))
model.add(Dropout(0.2))
model.add(Dense(units=64, activation='tanh'))
model.add(Dropout(0.2))
model.add(Dense(units=8, activation='softmax'))
model.summary()
```

Ek-3. GRU Kod Blokları

Görüntüden metin üreten fonksiyon

```
def metin_getir(resim_yolu, max_tokens=30):
    image = load_image(resim_yolu, size=img_size)

    image_batch = np.expand_dims(resim, axis=0)

    transfer_values = resim_model_transfer.predict(image_batch)

    decoder_input_data = np.zeros(shape=(1, max_tokens), dtype=np.int)

    token_int = token_start
    output_text = ''
    count_tokens = 0

    while token_int != token_endandcount_tokens<max_tokens:
        decoder_input_data[0, count_tokens] = token_int

        x_data = {'transfer_values_input': transfer_values, 'decoder_input': decoder_input_data}

        decoder_output = decoder_model.predict(x_data)

        token_onehot = decoder_output[0, count_tokens, :]
        token_int = np.argmax(token_onehot)

        sampled_word = tokenizer.token_to_word(token_int)
        output_text += " " + sampled_word
        count_tokens += 1

    plt.imshow(resim)
    plt.show()

    print(Metni getir:')
    print(output_text)
    print()
```

Ek-4. Sesi Metne Çeviren Kod Blokları

```
importspeech_recognition
ses_dinle = speec_recognition.Recognizer()
withspeech_recognition.AudioFile(‘ses_kaydi.wav’) as dosya:
ses_dosyasi = ses_dinle.listen(dosya)
try :
metin = ses_dosyasi = ses_dinle.recognize_google(ses_dinle, language = ‘tr’)
        print(‘ses dosyası:’)
        print(‘metin’)
except:
print(‘yeniden dene’)
```

Ek-5. Dil Tespiti Yapan Kod Blokları

SparkNLP için, kodlar https://sparknlp.org/sitesinden alınmıştır.

```
language_detector = LanguageDetectorDL.pretrained("ld_wiki_tatoeba_cnn_375", "xx")\
.setInputCols(["sentence"])\
.setOutputCol("language")
languagePipeline = Pipeline(stages=[documentAssembler, sentenceDetector, language_detector])
light_pipeline = LightPipeline(languagePipeline.fit(spark.createDataFrame([[""]]).toDF("text")))
result = light_pipeline.fullAnnotate("Spark NLP estunebibliothèque de traitement de texteopensourcepour
le traitementavancédulangagenaturelpourleslangages de programmation Python, Java et Scala.")
#program çıktısı
‘fr’ # kullanılan dilin Fransızca olduğunu tespit etmiştir.
```

Ek-6. Spark NLP alt düzey ve üst düzey özellikleri.

Alt düzey NLP özellikleri

Özellik	Spark NLP ile Spark Hizmeti (Azure Databricks, Azure Synapse Analytics, Azure HD Insight)	Azure Bilişsel Hizmetler
Cümle algılayıcısı	Evet	Hayır
Derin cümle algılayıcısı	Evet	Evet
Belirteç	Evet	Evet
N-gram oluşturucu	Evet	Hayır
Sözcük segmentasyonu	Evet	Evet
Stemmer	Evet	Hayır
Lemmatizer	Evet	Hayır
Konuşma bölümü etiketleme	Evet	Hayır
Bağımlılık ayrıştırıcısı	Evet	Hayır
Çeviri	Evet	Hayır
Stopword	Evet	Hayır
Yazım düzeltmesi	Evet	Hayır
Normalizer	Evet	Evet
Metin eşleştirici	Evet	Hayır
TF/IDF	Evet	Hayır
Normal ifade eşleştiricisi	Evet	Hayır
Tarih eşleştirici	Evet	Hayır
Öbekleyici	Evet	Hayır

Üst düzey NLP özellikleri

Özellik	Spark NLP ile Spark Hizmeti (Azure Databricks, Azure Synapse Analytics, Azure HD Insight)	Azure Bilişsel Hizmetler
Yazım denetimi	Evet	Hayır
Özetleme	Evet	Evet
Soru yanıtlama	Evet	Evet
Yaklaşım algılama	Evet	Evet
Duygu algılama	Evet	Fikir madenciliği destekler
Belirteç sınıflandırması	Evet	Evet, özel modeller aracılığıyla
Metin sınıflandırması	Evet	Evet, özel modeller aracılığıyla
Metin gösterimi	Evet	Hayır
NER	Evet	Evet
Varlık tanıma	Evet	Evet
Dil algılama	Evet	Evet
İngilizce dışında dilleri de destekler	Evet, 200'den fazla dili destekler	Evet, 97'den fazla dili destekler