

T.C.
FIRAT ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ

**BELEDİYELERE GELEN ŞİKAYET METİNLERİNİN
YAPAY ZEKA YÖNTEMLERİ İLE SINIFLANDIRILMASI**

Yasin GÜNAY

Tez Danışmanı
Prof. Dr. Ahmet Bedri ÖZER

BİTİRME TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

ELAZIĞ - 2024

T.C.
FIRAT ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ

BELEDİYELERE GELEN ŞİKAYET METİNLERİNİN
YAPAY ZEKA YÖNTEMLERİ İLE SINIFLANDIRILMASI

Yasin GÜNAY

BİTİRME TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

Bu bitirme tezi/...../2024 tarihinde, aşağıda belirtilen jüri tarafından oybirliği/oyçokluğu ile başarılı/başarısız olarak değerlendirilmiştir.

(imza)

(imza)

(imza)

Prof. Dr. Ahmet Bedri ÖZER

(imza)

ÖZGÜNLÜK BİLDİRİMİ

Bu çalışmada, başka kaynaklardan yapılan tüm alıntılar, ilgili kaynaklar referans gösterilerek açıkça belirtildiğini, alıntılar dışındaki bölümlerin, özellikle projenin ana konusunu oluşturan teorik çalışmaların ve yazılım/donanımın benim tarafımdan yapıldığını bildiririm.

Fırat Üniversitesi

.../.../2024

Bilgisayar Mühendisliği

23119 Elazığ

Öğrencilerin Adı Soyadı

Yasin GÜNAY

BENZERLİK BİLDİRİMİ

YASİN GÜNAY

ORJİNALLİK RAPORU

% 13	% 11	% 5	% 7
BENZERLİK ENDEKSİ	İNTERNET KAYNAKLARI	YAYINLAR	ÖĞRENCİ ÖDEVLERİ

BİRİNCİL KAYNAKLAR

1	dergipark.org.tr İnternet Kaynağı	% 5
2	Submitted to Fırat Üniversitesi Öğrenci Ödevi	% 2
3	Submitted to Istanbul University Öğrenci Ödevi	% 1
4	Submitted to Sivas Cumhuriyet Üniversitesi Öğrenci Ödevi	% 1
5	Submitted to Istanbul Aydın University Öğrenci Ödevi	<% 1
6	Submitted to Anadolu University Öğrenci Ödevi	<% 1
7	Submitted to Bahcesehir University Öğrenci Ödevi	<% 1
8	ebin.pub İnternet Kaynağı	<% 1
9	Cura, Esra. "Katlama Ve Giberellik Asit Uygulamalarının Sarımsak (Allium sativum L.) Tohumlarında Dormansinin Kırılması ve	<% 1

TEŞEKKÜR

Bu çalışmam, Fırat Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümü'nde, Sayın Prof. Dr. Ahmet Bedri ÖZER'in yönlendirmesi ve gözetimi altında hazırlanmıştır. Projenin her aşamasında desteklerini esirgemeyen hocam, Sayın Prof. Dr. Ahmet Bedri ÖZER'e en içten duygularıyla teşekkür ederim.

Yasin GÜNAY

İÇİNDEKİLER

	<u>Sayfa No</u>
ÖZGÜNLÜK BİLDİRİMİ	I
BENZERLİK BİLDİRİMİ	II
TEŞEKKÜR.....	III
İÇİNDEKİLER	IV
ŞEKİLLER LİSTESİ	V
KISALTMA LİSTESİ.....	VI
ÖZET	VII
ABSTRACT.....	VIII
1. GİRİŞ.....	1
2. TEORİK ALTYAPI.....	2
2.1. Yapay Zeka	2
2.2. Yapay Sinir Ağları	3
2.3. Derin Öğrenme.....	3
2.4. Python	5
2.5. Doğal Dil İşleme	5
2.6. Metin Sınıflandırma	6
2.7. BERT Modeli	8
2.8. Transformer Mimarisi	9
2.9 Fine-Tuning (İnce Ayar)	10
3. UYGULAMA	12
3.1. Uygulamanın Temel Özellikleri	12
3.2. Gerekli Kütüphane Ve Araçların Yüklenmesi.....	12
3.3. Veri Seti.....	13
3.4. Pretrained Model Ve Tokenizer'ın Yüklenmesi.....	14
3.5. Veri Setlerinin Model Eğitimine Hazırlanması	14
3.6. Değerlendirme Metriği	15
3.7. Eğitim Parametrelerinin Ayarlanması	15
3.8. Eğitim	15
3.9. Arayüz	16
4. SONUÇLAR	18
5. KAYNAKLAR.....	19
6. ÖZGEÇMİŞ.....	21

ŞEKİLLER LİSTESİ

Sayfa No

Şekil 2.1.	Yapay Zeka'nın Kronolojik Tarihi	2
Şekil 2.2.	Sinir Hücresine Ait Matematiksel Model	3
Şekil 2.3.	YSA ve CNN Topolojileri	4
Şekil 2.4.	Derin Öğrenme Algoritmalarının Yapay Zekadaki Konumu	5
Şekil 2.5.	Doğal Dil İşleme Adımları	6
Şekil 2.6.	Metin Sınıflandırma Adımları	8
Şekil 2.7.	Popüler Transformer Model Sürümlerine Ait Zaman Çizelgesi.....	9
Şekil 2.8.	Pretraining Eğitim Aşamaları	10
Şekil 2.9.	Pretrained Modelin Fine-tuning Edilme Aşamaları.....	11
Şekil 3.1.	Paket Yükleyici	12
Şekil 3.2.	Kütüphanelerin Yüklenmesi	12
Şekil 3.3.	Veri Setinden Veri Örnekleri.....	13
Şekil 3.4.	Veri Setlerinin Yüklenmesi	13
Şekil 3.5.	Pretrained Model Ve Tokenizer	14
Şekil 3.6.	Veri Setlerinin Ön İşlenmesi	14
Şekil 3.7.	Değerlendirme Metriği	15
Şekil 3.8.	Parametre Değerleri.....	15
Şekil 3.9.	Eğitim Döngüsü.....	16
Şekil 3.10.	Şikayet Arayüzü	17
Şekil 3.11.	Sonuç Arayüzü	17

KISALTMA LİSTESİ

- AI** : Artifical Intelligence (Yapay Zekâ)
- BART** : Bidirectional Autoregressive Transformer (Çift Yönlü Otoresif Transformatör)
- BERT** : Bidirectional Encoder Representations from Transformer (Transformatörlerden Çift Yönlü Kodlayıcı Temsilleri)
- CSV** : Comma Seperated Values (Virgülle Ayrılmış Değerler)
- GPT** : Generative Pre-trained Transformer (Üretken Önceden Eğitilmiş Transformatör)
- NLP** : Natural Language Processing (Doğal Dil İşleme)
- TPU** : Tensor Processing Unit (Tensör İşleme Birimi)
- CNN** : Convolutional Neural Network
- YSA** : Yapay Sinir Ağları
- DL** : Deep Learning (Derin Öğrenme)

ÖZET

Günümüzde yapay zekanın gösterdiği gelişmeler doğrultusunda, farklı sektörler için birçok farklı yapay zeka çözümü hayatlarımıza girmektedir. Son yıllarda, özellikle Doğal Dil İşleme konusunda büyük gelişmeler yaşanmaktadır. Bu tez kapsamında ortaya konulmuş olan uygulama ve araştırma projesi, Doğal Dil İşleme yöntem ve tekniklerini kullanarak, belediyeçilik alanında bir metin sınıflandırma çözümü önermektedir. Belediyelerin, vatandaşların istek ve şikayetleri ile etkili bir biçimde ilgilenebilmesi büyük bir önem arz etmektedir. Gün geçtikçe artan nüfus ve sistemlerin karmaşıklaşması bu durumu daha zor bir hale getirmektedir. Bitirme projesi kapsamında önerilen çözüm, vatandaşların belediyelere yazmış oldukları şikayet metinlerini, yapay zeka yöntemleri ile ilgili birim bazında otomatik bir şekilde sınıflandırmaktır. Bu sınıflandırma işlemi, vatandaşların şikayetçi oldukları konular ile doğrudan ilgilenebilecek birim ile etkili bir iletişim geliştirmeleri açısından büyük fayda sağlayacaktır. Diğer taraftan, belediyelerin verdiği hizmetlerin kalitesinde ve hızında iyileştirmelere kapı açacaktır.

Bu proje kapsamında eğitilmiş olan model, 8 farklı sınıftan oluşmaktadır. Veriler, vatandaşların çeşitli belediyelere internet üzerinden yazdıkları şikayet metinlerinden oluşmaktadır. Transformer mimarisinin getirmiş olduğu dikkat mekanizmasının kullanımı bu modelin başarısı için oldukça önemli bir rol oynamıştır. Model eğitimi, BERT tabanlı bir pretrained sınıflandırma modeli kullanılarak fine-tuning tekniği ile gerçekleştirilmiştir. Modelin kullanımına örnek olması açısından bir web arayüzü hazırlanmıştır. Bu tez, belediyeçilik alanında çözüm sunması hedeflenen bir metin sınıflandırma modelinin eğitimini ve incelenmesini içermektedir.

Anahtar Kelimeler: Yapay Zeka , Metin Sınıflandırma , BERT , Transformer , Doğal Dil İşleme

ABSTRACT

Today, in line with the developments in artificial intelligence, many different artificial intelligence solutions for different sectors are entering our lives. In recent years, there have been great developments especially in Natural Language Processing. The application and research project presented in this thesis proposes a text classification solution in the field of municipalities using Natural Language Processing methods and techniques. It is of great importance for municipalities to deal effectively with citizens' requests and complaints. Increasing population and complexity of the systems make this situation more difficult. The solution proposed within the scope of the graduation project is to automatically classify the complaint texts written by citizens to the municipalities on the basis of the relevant unit with artificial intelligence methods. This classification process will be of great benefit for citizens to develop effective communication with the unit that can directly deal with the issues they complain about. On the other hand, it will open the door to improvements in the quality and speed of services provided by municipalities.

The model trained in this project consists of 8 different classes. The data consists of complaints written by citizens to various municipalities over the internet. The use of the attention mechanism introduced by the Transformer architecture played a crucial role in the success of this model. The model training was performed by fine-tuning technique using a BERT-based pretrained classification model. A web interface was prepared as an example for the use of the model. This thesis includes the training and investigation of a text classification model that aims to provide a solution in the field of municipality.

Keywords: Artificial Intelligence , Text Classification , BERT , Transformer , Natural Language Processing

1. GİRİŞ

Bilgi teknolojileri alanında bir devrime tanıklık ettiğimiz bu günlerde, şüphesiz ki bu devrimin en önemli unsurlarından biri yapay zeka teknolojileri olmuştur. Yapay zeka teknolojilerinin özellikle son yıllarda kullanım oranları ve tanınırlığı ciddi ölçüde artmıştır. Bu durumun önemli sebeplerinden biri de NLP alanında devrimsel boyutta yeniliklerin gerçekleşmiş olmasıdır. NLP, insanların kullandığı doğal dili anlama, yorumlama, mantıksal çıkarımlar yapma, diller arası çeviriler yapma ve metin üretme gibi konular üzerine yoğunlaşan bir yapay zeka alanıdır.

Metin sınıflandırma problemleri, doğal dil işlemenin önemli bir alt dalıdır. Doğal dilde yazılmış metinlerin, önceden belirlenmiş kategorilere veya sınıflara ayrılmasını amaçlar. Metinlerin manuel olarak etiketlendiği veri setleri ile eğitilen bu modeller, kendisine gelen yeni metni işleyerek hangi sınıfa ait olduğunu tahmin eder. Geçmişten bugüne metin sınıflandırma problemlerinin çözümü için birçok farklı yaklaşım geliştirilmiştir. Bu yaklaşımlar, Naive Bayes, Destek Vektör Makineleri, Lojistik Regresyon ve Derin Öğrenme Yöntemleri gibi farklı teknikler ortaya koyan yaklaşımlardır. Özellikle son yıllarda derin öğrenmeye dayalı, yapay sinir ağlarını kullanan Transformer gibi yaklaşımlar oldukça başarılı sonuçlar vermektedir.

Transformer mimarisi doğal dil işleme alanında çığır açan bir mimaridir. Google tarafından 2017 yılında tanıtılmıştır. Transformer mimarisini ön plana çıkaran en önemli bileşen, dikkat mekanizması olmuştur. Bu mekanizma sayesinde, kelimelerin metin içerisindeki bağlamsal anlamı daha etkili bir biçimde anlaşılır. Transformer mimarisi kullanılarak, birçok dil modeli geliştirilmiştir. Bunların arasında Google tarafından geliştirilen BERT modeli metin sınıflandırma problemleri için oldukça başarılı sonuçlar vermektedir.

Bu tez kapsamında, BERT tabanlı bir metin sınıflandırma modelinin, ince ayar yöntemi ile özel amaçlı bir uygulama için tekrar eğitilmesi ve kullanılması incelenmiştir. Eğitilen sınıflandırma modelinin görevi belediyelere yapılan istek ve şikayet metinlerini ilgili olduğu birim bazında sınıflandırmaktır. Python dilinden ve bu dilin sağladığı Transformers kütüphanesinden proje kapsamında yoğun bir biçimde istifade edilmiştir. Ön eğitilmiş model kullanılarak, özel amaçlı veriler, sınıflar ve parametreler ile gerçekleştirilen bu eğitim yöntemi Fine-Tuning olarak adlandırılır. Bu tezin temel hedefi özel amaçlı eğitilmiş bir sınıflandırma modeli üzerinden mimari, model ve tekniklerin kullanımlarını ve elde edilen sonuçları incelemektir.

2. TEORİK ALTYAPI

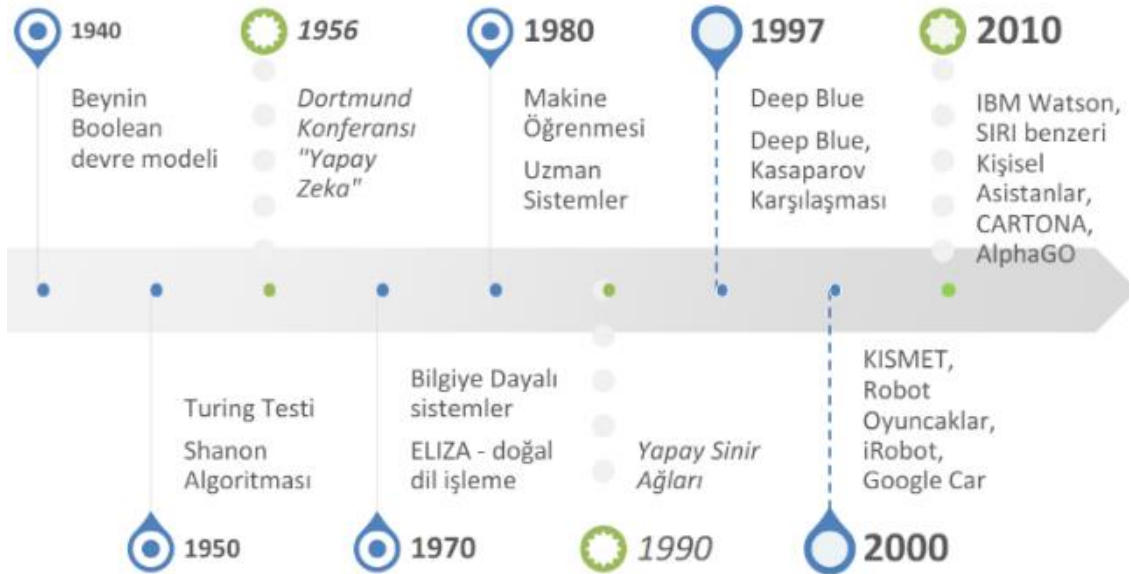
2.1. Yapay Zeka

1956 yılında John McCarthy, Marvin L. Minsky, Nathaniel Rochester ve Claude E. Shannon, Dortmund Konferansı'nda yapay zekâ kavramını ortaya atan bir öneri mektubu sunmuşlardır. Ancak bu kavramın yaratıcısı olarak John McCarthy kabul edilmektedir [1]. McCarthy zekâyı “gerçek dünyada hedeflere ulaşma yeteneğinin hesaplamaya dayalı bir parçası” olarak tanımlamaktadır. İnsanlar, birçok hayvan ve bazı makineler çeşitli türlerde ve derecelerde zekâ sergilerler” şeklinde tanımlarken, yapay zekâyı ‘insan benzeri zeki makineler, özellikle de zeki bilgisayar programları yaratma bilimi ve mühendisliği’ olarak tanımlamaktadır [2].

Örneğin, Slage yapay zekayı “sezgisel programlama” olarak tanımlamıştır. Benzer şekilde Axe da yapay zekayı karmaşık problemleri çözebilen ve sadece planlanmış zorluklara değil yeni durumlara da cevap üretebilen akıllı programlar olarak kabul etmiştir [3]. Yapay zekâ üzerine yaptığı araştırmalarla literatürde tanınan bir diğer isim olan Nils Nilsson'a göre yapay zekâ, doğal zekâyı taklit etmeye çalışan bir hipotezdir [4].

Görülebileceği gibi, çeşitli tanımlar olsa da, hepsi yapay zekanın iki kavram üzerine kurulabileceği konusunda hemfikirdir: “akıllı programlama” ve “insansı tepkiler”.

Günümüzde yapay zeka uygulamaları ve araştırmaları önemli bir gelişim göstermektedir. Üretilen yapay zeka çözümleri, birçok alanda süreçleri daha verimli bir hale getirmiştir. Yapay zeka çözümlerinin gelecekte daha da yaygınlaşması ve önemini artırması beklenmektedir.

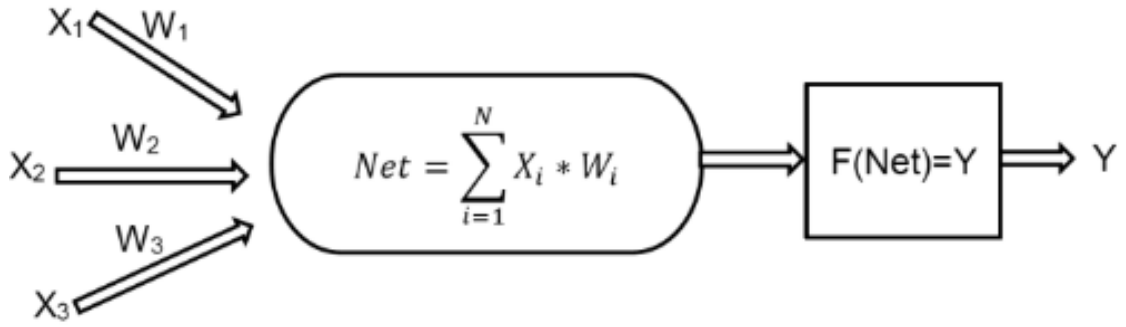


Şekil 2.1. Yapay Zeka'nın Kronolojik Tarihi

2.2. Yapay Sinir Ağları

Yapay sinir ağları (YSA), insan beyninde bulunanlara benzer sinir hücrelerinin gruplandırılmasıyla oluşturulur. En temel insan özelliklerinden biri olan öğrenme yeteneğine sahiptir. Her işlem ögesi, ağırlıklar kullanılarak diğerleriyle ilişkilendirilir. Öğrenme, ağın en doğru sonuçları elde etmek için bu ağırlıkları ayarlama sürecidir. Sonuç olarak, yeni senaryoları öğrenmek için girdilere dayalı olarak ağırlıklar öğretilir [5].

Bir sinir ağı yapısı genellikle üç temel bileşenden oluşur: giriş katmanı (giriş değerleri veya nitelikler), gizli katmanlar ve çıkış katmanı. Giriş katmanı, başka bir sinir ağından alınan çıkış değerlerini veya işlenecek nesnenin özelliklerini depolar. Gizli katmanlar bu nitelikleri ve belirli ağırlık değerlerini kullanarak bir özellik haritası oluşturur. Bu öz nitelikler, aktivasyon fonksiyonları aracılığıyla çıktı katmanında anlamlı bir sonuç üretir [6]. Aşağıdaki şekilde gösterilen model incelendiğinde, x girişleri w ağırlıkları ile çarpılır ve daha sonra toplam fonksiyonu aracılığıyla gönderilir. Hesaplanan toplam, transfer fonksiyonu $F(\text{net})$ kullanılarak çıkışa aktarılır [7].



Şekil 2.2. Sinir Hücresine Ait Matematiksel Model

2.3. Derin Öğrenme

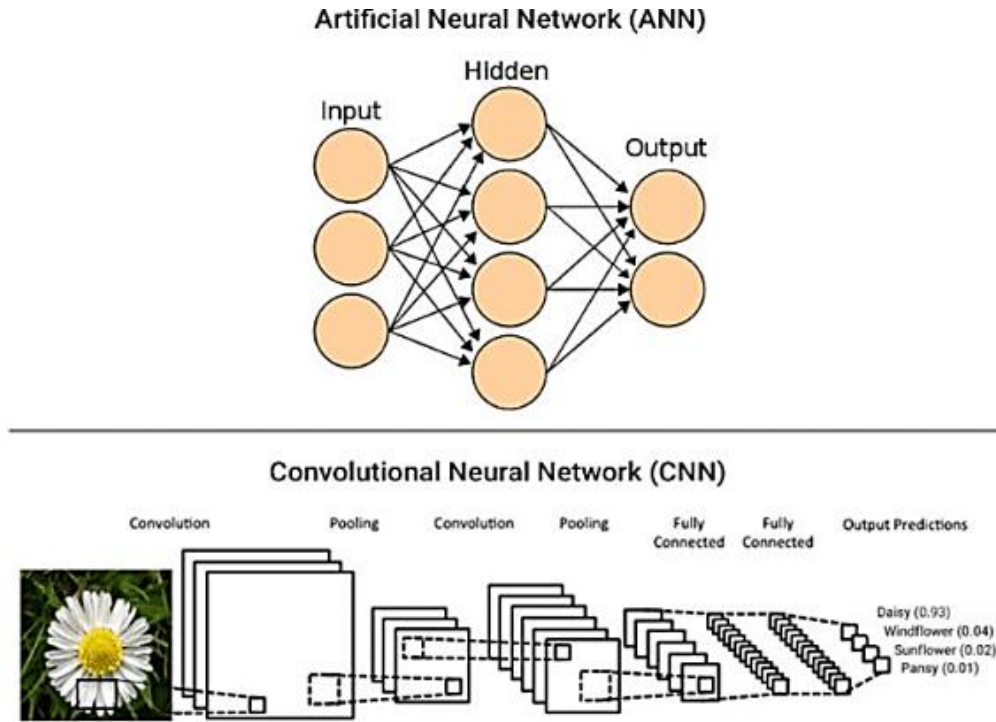
Derin öğrenme (DL), çeşitli soyutlama düzeylerinde veri temsillerini öğrenmek için birkaç işlem katmanına sahip hesaplama modellerini mümkün kılar. Bu yöntemler, konuşma tanıma, görsel nesne tanıma, nesne algılama ve ilaç geliştirme ve genomik dahil olmak üzere çeşitli diğer alanlarda en son teknolojiyi önemli ölçüde geliştirmiştir. Derin öğrenme, bir makinenin bir önceki katmandaki temsile dayalı olarak her katmandaki temsili hesaplamak için gereken dahili parametrelerini nasıl değiştirmesi gerektiğini belirten geri yayılım algoritmasını kullanarak büyük veri kümelerinin karmaşık yapısını araştırır [8].

Derin öğrenme, klasik makine öğreniminde veri temsillerini manuel olarak oluşturma yükünü hafifletmiştir. Derin öğrenme, verileri birbirini izleyen katmanlarda işleyerek giderek daha kullanışlı temsiller ortaya çıkarır. Derin öğrenme, özellik çıkarma yükünü taşıdığı için

popüleritesini artırmıştır. Derin öğrenme çalışmaları giderek daha popüler hale gelmektedir [9]. Derin öğrenme çalışmaları, devasa veri setlerinin artan kullanılabilirliği ve bilgisayar donanımının iyileştirilmesi ile desteklenmektedir.

Derin öğrenmede derinlik, sonraki katmanları ifade eder [10]. Derin öğrenme ve derin sinir ağları çok katmanlı yapay sinir ağları olarak da temsil edilebilir. YSA dönüşümleri ileri beslemeli sinir ağları, geri beslemeli sinir ağları veya çok katmanlı sinir ağları olarak sınıflandırılabilir. Konvolüsyonel Sinir Ağı (CNN) en sık kullanılan derin sinir ağlarından biridir. Adını, konvolüsyon olarak bilinen matrisler arasındaki matematiksel doğrusal eylemden alır. CNN, konvolüsyonel katman, doğrusal olmayan katman, havuz katmanı ve tam bağlı katman dahil olmak üzere çeşitli katmanlardan oluşur [11].

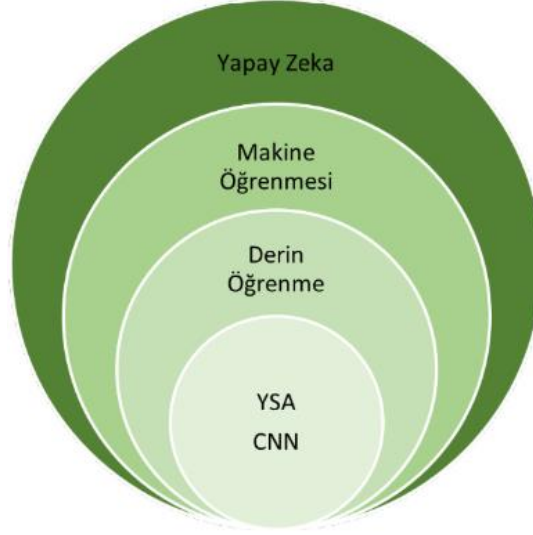
CNN'ler özellikle bilgisayarla görme alanında önemli başarılar kazanmıştır [12]. Aşağıdaki şekilde bir YSA ve bir CNN'in topolojik yapıları gösterilmektedir. Bir sınıflandırma görevinde, bir CNN'in çıktıları girdinin ait olduğu sınıfları temsil eder. Bir CNN'in eğitim aşaması tipik olarak zaman alıcı ve kaynak yoğundur. Eğitimi takip eden tahmin aşaması hızlıdır ve mütevazı kaynaklar gerektirir [13].



Şekil 2.3. YSA ve CNN Topolojileri

Makine öğrenimi yapay zekanın bir alt kümesidir. YSA, makine öğreniminin bir türü olarak düşünülebilir. CNN'ler YSA'ların evrimleşmiş halidir ve Derin Öğrenme bağlamında da

kullanılmaktadır. Derin öğrenme de makine öğrenmesinden ayırt edilebilen temel YSA ve CNN yapılarına sahip bir yapay zeka alanı olarak düşünülebilir [14].



Şekil 2.4. Derin Öğrenme Algoritmalarının Yapay Zekadaki Konumu

2.4. Python

Guido van Rossum Python'un ilk sürümünü 1991 yılında tanıtmıştır. Yorumlanmış ve dinamik bir dil olan Python, çoğunlukla prosedürel ve nesne yönelimli programlama paradigmlarını desteklemekle birlikte fonksiyonel programlamayı da desteklemektedir. Python, göreceli basitliği ve kapsamlı standart kütüphanesinin bir sonucu olarak popülaritesini artırmıştır ve artık şirketler de dahil olmak üzere geniş bir kullanıcı tabanına sahiptir [15].

Python'ın en önemli kullanım alanlarının başında veri bilimi ve yapay zeka uygulamaları gelmektedir. Bu alanlarda yapılan çalışmalar için çok büyük bir kütüphane desteğine ve topluluklara sahiptir. Karmaşık modellerin eğitimi ve büyük veriler ile çalışmak Python'ın sağladığı kütüphane ve araçlarla daha hızlı ve basit bir hale gelmiştir. Proje kapsamında transformers, numpy, datasets gibi Python kütüphanelerinden yararlanılmıştır.

2.5. Doğal Dil İşleme

Doğal dil işleme, dijital cihazların sosyal medya internet ortamındaki verilerden ilgili ve istenen bilgileri çıkarmasına olanak tanıyan yapay zeka destekli bir teknolojidir. Akıllı uygulamalar doğal dil işlemeden yararlanır [16]. Doğal dil işlemede ön işleme süreçleri arasında kelime normalleştirme, gürültü azaltma ve nesne standartlaştırma yer alır. Kelime normalleştirme yaklaşımı, aynı kelime kökünden türetilen terimleri normalleştirir. Gürültü azaltma tekniği,

cümlelerdeki bağlaç terimlerini tespit eder ve belirsizliği ortadan kaldırır. Nesne standardizasyonu, sosyal veya dijital ortamlarda kullanılan kısaltmaların anlamını belirlemeye yönelik bir stratejidir. Ön işleme prosedürlerinin ardından, doğal dil işleme yöntemi, kelimeleri oluşturan cümleler arasındaki özne-nesne-yüklem bağlantısını bulan varlık çıkarma işlemini gerçekleştirir. Doğal dil işleme, cümle sıklığı ve kelime sayısı gibi bilgileri de belirleyebilir [17].



Şekil 2.5. Doğal Dil İşleme Adımları

Doğal dil işlemenin temel kavramları arasında söz dizimi, anlamsal analiz ve anlam belirsizliği bulunur. Sözdizimi, dillerdeki cümle yapısını yöneten ilke ve kuralların yanı sıra cümle esnekliğini inceleyen dilbilim disiplini. Semantik yani anlamsal analiz, kelime ve kelime gruplarının anlamlarının yanı sıra aynı kelimelerin çeşitli yorumlarını ve aktarıldıkları bağlamı inceleyen bir dilbilim disiplini. Anlam ayrımı yani belirsizliği, bir sözcüğün her durumda ne anlama geldiğini belirleme sürecidir.

Günümüzde doğal dil işleme alanında karşılaşılan en ciddi zorluklardan bazılarına yönelik yeni çözümler sürekli olarak araştırılmaktadır. Doğal Dil İşlemenin temel çalışma alanları arasında; metin sınıflandırma, metin özetleme, duygu analizi, benzerlik analizi, diller arası çeviri, bilgi çıkarma, metin oluşturma, soru cevaplama gibi konular bulunmaktadır.

2.6. Metin Sınıflandırma

Belgeleri önceden belirlenmiş kategorilere ayırma tekniği metin sınıflandırma olarak bilinir [18]. 1960'larda başlayan metin madenciliği, büyük miktarda metinsel veriden bilgi çıkarma sürecidir. Önemi 2000'li yıllardan bu yana artmıştır [19].

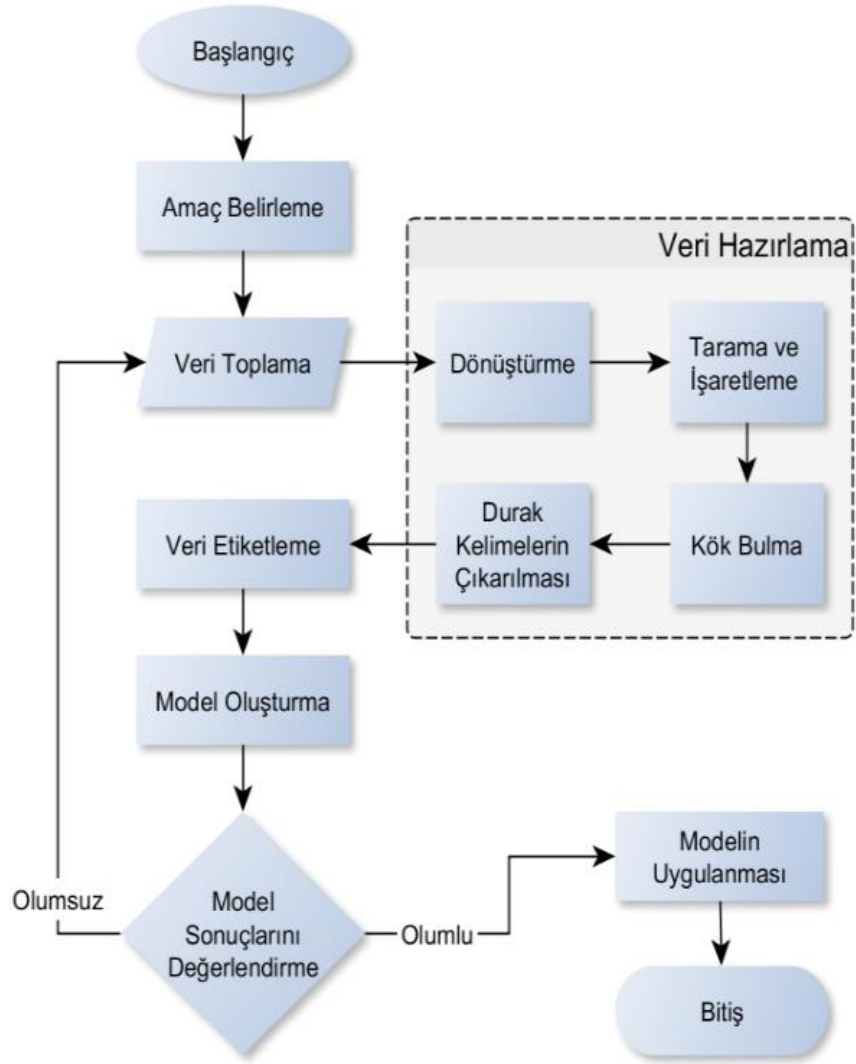
Metin sınıflandırma için çok sayıda kullanım alanı vardır. Metin sınıflandırma uygulamaları arasında metin filtreleme, spam filtreleme, metinlerden yazarı ve metin dilini belirleme, kategoriler elle oluşturulduğunda kütüphane organizasyonu gibi ortamlarda kategori atamasına yardımcı olma ve çok daha fazlası yer alır.

Metin sınıflandırma konularındaki temel zorluk, başarı oranını düşüren veri uzayının genişliğidir. Bu sorunu ele almak için, özellik seçimi genellikle literatür araştırmalarında bir ön işleme adımı olarak gerçekleştirilir. Boyutluluğu öncelik sırasına göre azaltarak, özellik seçimi işlemi hesaplamayı hızlandırır ve başarı oranlarını artırır. Kullanılan verilerin büyük boyutluluğu sınıflandırma sürecini zorlaştırdığından, boyut azaltma çok önemlidir [20].

Sınıflandırma, yeni bir nesnenin özelliklerine bir açıklama verme ve onu bir dizi belirlenmiş sınıftan birine yerleştirme işlemidir [21]. Metin sınıflandırma günümüzde birçok farklı alanda kullanılmaktadır. Bunlar şu şekilde özetlenebilir ancak bunlarla sınırlı değildir: belgelere sınıf atamak, sözcük analizi görevlerine yardımcı olmak (sözcük anlam ayrımı gibi), metni sınıf tabanlı işleme mekanizmalarına yönlendirmek, doğal dil işleme sistemlerinin bir parçası olarak alakasız metin ve metin parçalarını filtrelemek, sınırlı bilgi alma biçimlerini etkinleştirmek [22].

Bugüne kadar metin sınıflandırmada çeşitli teknikler kullanılmıştır. k-NN sınıflandırma (k-NN) [23-25], Naïve Bayes olasılıksal sınıflandırma [23,24], karar ağaçları [23,24], yapay sinir ağları [23,24] ve destek vektör makineleri bu tekniklerden birkaçıdır. Ayrıca, birçok yaklaşımın bir araya getirildiği ve sonuçların karşılaştırıldığı araştırmalar da yapılmıştır [26-30].

Metin sınıflandırma süreci eğitim ve sınıflandırma aşamalarından oluşur. Eğitim aşamasında, sınıfı bilinen veri tabanından rastgele seçilen bir veri alt kümesi bir model üretmek için eğitilir. Öğrenme setinin bir parçası olmayan veri tabanından seçilen veriler sınıflandırma işleminde kullanılır [31]. Aşağıdaki şekilde metin sınıflandırma işleminin adımları gösterilmektedir.



Şekil 2.6. Metin Sınıflandırma Adımları

2.7. BERT Modeli

Doğal dil işleme için, Transformatörlerden Çift Yönlü Kodlayıcı Temsilleri anlamına gelen BERT, bir makine öğrenimi (ML) modelidir. 2018 yılında Google AI Language araştırmacıları tarafından geliştirilmiştir.

Geleneksel olarak, her bir benzersiz NLP görevini çözmek için farklı modeller geliştirilmiştir. BERT, en sık karşılaşılan 11'den fazla NLP görevini çözerek (ve bunu daha önceki modellerden daha etkili bir şekilde yaparak) NLP alanını dönüştürdü ve tüm NLP işlemlerinin ustası haline geldi.

BERT, aşağıda görüldüğü üzere çok çeşitli görevlere uyarlanabilir:

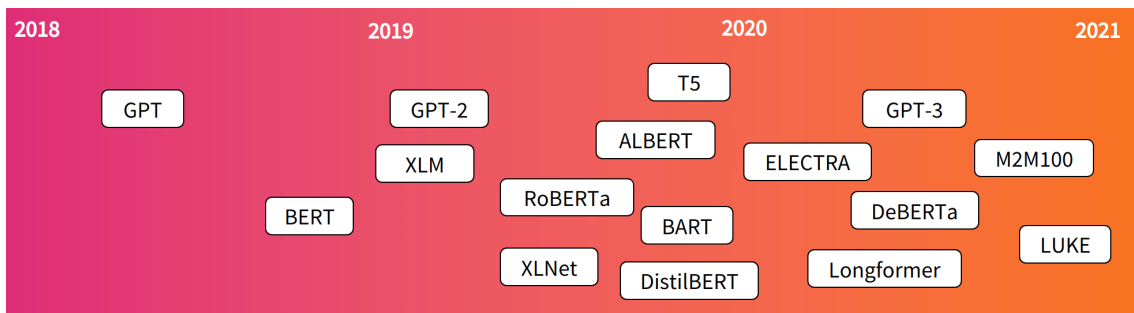
- Duygu Analizi (Sentiment Analysis)

- Soru Cevaplama (Question Answering)
- Metin Tahmini (Text Prediction)
- Metin Oluşturma (Text Generation)
- Özetleme (Summarization)
- Çokanlamlılık Çözümlemesi (Polysemy Resolution)

BERT'in çalışma mekanizması sürdürülebilir performansı açısından 3.3 milyar kelimeden oluşan geniş bir veri kümesine dayanmaktadır. BERT özellikle Google BooksCorpus (yaklaşık 800 milyon kelime) ve Wikipedia (yaklaşık 2,5 milyar kelime) üzerinde eğitilmiştir. Bu oldukça büyük bilgilendirici veri kümeleri, BERT'in hem İngilizce diline hem de daha geniş bir dünyaya ilişkin derin anlayışını geliştirmiştir. Bu büyüklükteki bir veri kümesi üzerinde eğitim yapmak biraz zaman almaktadır. Yenilikçi Transformer mimarisi BERT'in eğitilmesini sağlarken TPU'lar (Tensör İşleme Birimleri, Google'ın özellikle devasa makine öğrenimi modelleri için tasarlanmış özel devresi) bunu daha hızlı yapmasını sağlamıştır. BERT, 64 TPU tarafından dört gün boyunca eğitilmiştir [32].

2.8. Transformer Mimarisi

Transformer mimarisi, makine öğrenimi eğitiminin inanılmaz derecede etkili bir şekilde paralelleştirilmesine olanak tanır. Böylece devasa paralelleştirme, BERT'in nispeten kısa bir sürede büyük hacimli veriler üzerinde eğitilmesini sağlar. Transformatörler bir dikkat mekanizması kullanarak kelime ilişkilerini fark eder.



Şekil 2.7. Popüler Transformer Model Sürümlerine Ait Zaman Çizelgesi

Haziran 2018: İlk ön eğitilmiş Transformer modeli olan GPT kullanılarak çeşitli NLP görevlerinde ince ayar yapılarak son teknoloji ürünü sonuçlar elde edildi.

Ekim 2018: BERT, daha doğru cümle özetleri üretmek amacıyla oldukça büyük bir ön eğitilmiş modeldir.

Şubat 2019: GPT-2, ilk başta ahlaki sorunlar nedeniyle kamuoyundan saklanan GPT'nin daha büyük ve daha iyi bir versiyonudur.

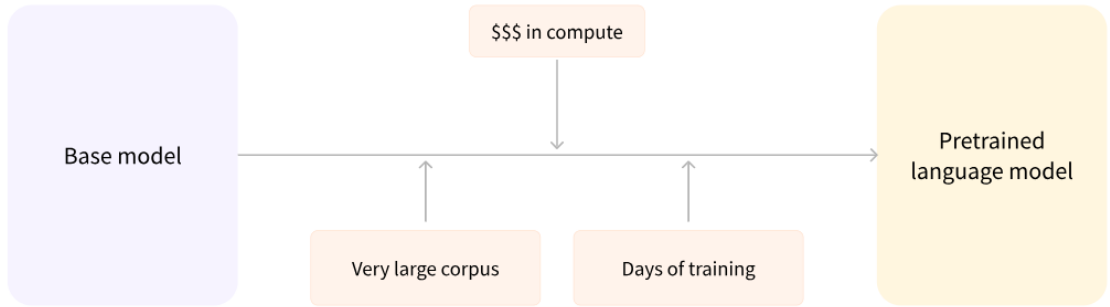
Ekim 2019: DistilBERT, BERT'in işlevselliğinin %97'sini korurken bellekte %60 daha hızlı ve %40 daha hafif olan azaltılmış bir BERT sürümüdür.

Ekim 2019: İki büyük ön eğitilmiş model olan BART ve T5, Transformer modeliyle aynı mimariyi ilk kullanan modellerdir.

GPT, BERT, BART, T5 ve diğerleri dahil olmak üzere yukarıda bahsedilen Transformer modellerinin her biri dil modeli eğitiminden geçmiştir. Bu, önemli miktarda işlenmemiş metin kullanarak kendi kendine denetimli eğitimden geçtiklerini gösterir. Kendi kendine denetimli öğrenmede, eğitim hedefi modelin girdilerine göre otomatik olarak hesaplanır. Bu, veri etiketlemenin insan müdahalesi gerektirmediği anlamına gelir [33].

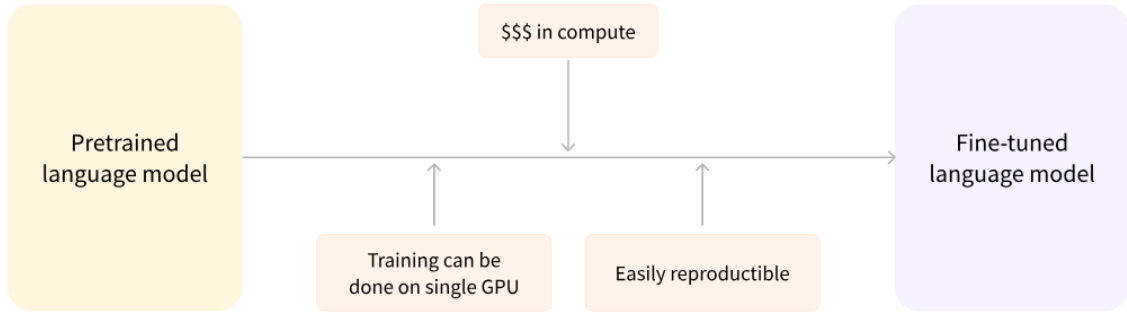
2.9 Fine-Tuning (İnce Ayar)

Ön eğitim, ağırlıkların rastgele başlatılmasıyla ve geçmiş bilgi olmadan başlayarak bir modeli sıfırdan eğitme sürecidir.



Şekil 2.8. Pretraining Eğitim Aşamaları

Ön eğitim genellikle çok büyük hacimlerde veri kullanılarak gerçekleştirilir. Bu nedenle, eğitim birkaç hafta sürebilir ve çok büyük bir veri kümesi gerektirir. Bunun aksine fine-tune, önceden eğitilmiş modellerin ardından gerçekleşen eğitimi ifade eder. Önceden eğitilmiş bir dil modeli elde etmek fine-tuning işleminin ilk adımındır. Ardından, özel bir veri kümesi kullanarak model daha fazla eğitilir.



Şekil 2.9. Pretrained Modelin Fine-tuning Edilme Aşamaları

Böylece, bir modelin fine-tune edilmesine ilişkin zaman, veri, mali ve çevresel maliyetler azalmaktadır. Eğitim, tam bir ön eğitime göre daha az kısıtlayıcı olduğundan ötürü çeşitli fine-tune stratejileri üzerinde yineleme yapmak da daha hızlı ve basit olacaktır. Ayrıca bir modeli eğitirken sıfırdan başlamak yerine (büyük miktarda veri yoksa) eldeki göreve mümkün olduğunca yakın olan önceden eğitilmiş bir modeli kullanmak ve onu iyileştirmek nispeten daha etkilidir [34].

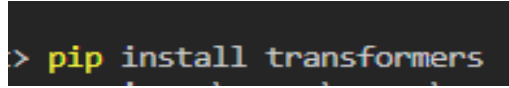
3. UYGULAMA

3.1. Uygulamanın Temel Özellikleri

Proje kapsamında gerçekleştirilmiş olan uygulama, BERT tabanlı bir ön eğitilmiş (pretrained) modelin, ince ayar yöntemi ile özel bir sınıflandırma görevi için eğitilmesine dayanmaktadır. Önceden belirlenmiş 8 farklı sınıf etiketi ile etiketlenen toplam 2698 adet metin verisi üzerinde eğitilmiştir. Belediyelere yapılan yazılı şikayetlerin, ilgili birim bazında sınıflandırılması problemini ele almaktadır. Modelin temel özelliklerinden, kurulumundan, eğitiminden ve çalışma prensiplerinden aşağıdaki bölümlerde detaylıca bahsedilmiştir.

3.2. Gerekli Kütüphane Ve Araçların Yüklenmesi

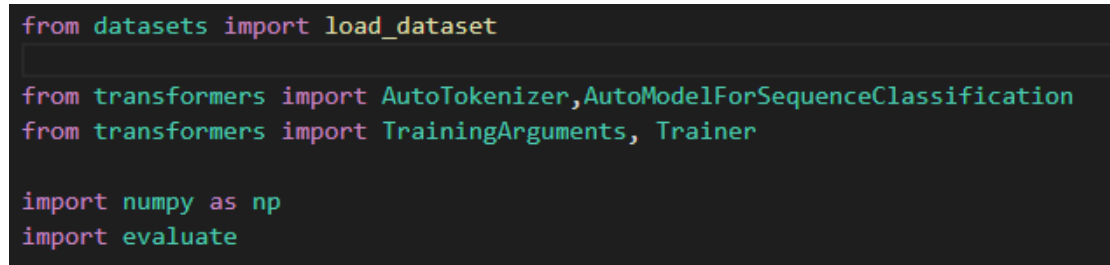
Proje kapsamında Python programlama diline ait, yapay zeka ve doğal dil işleme konusunda sıkça kullanılan bazı kütüphaneler, sınıflar ve fonksiyonlar kullanılmıştır. Bu araçların proje içerisinde kullanılabilmesi için öncelikle yüklenmesi, ardından import komutu ile proje koduna dahil edilmesi gerekmektedir. Yükleme işlemi terminal kullanılarak pip paket yükleyici ile gerçekleştirilmiştir.



```
> pip install transformers
```

Şekil 3.1. Paket Yükleyici

Gerekli kütüphaneler yüklendikten sonra, gerekli sınıf ve fonksiyonlar bu kütüphanelerden import komutu ile proje koduna dahil edilmiştir.



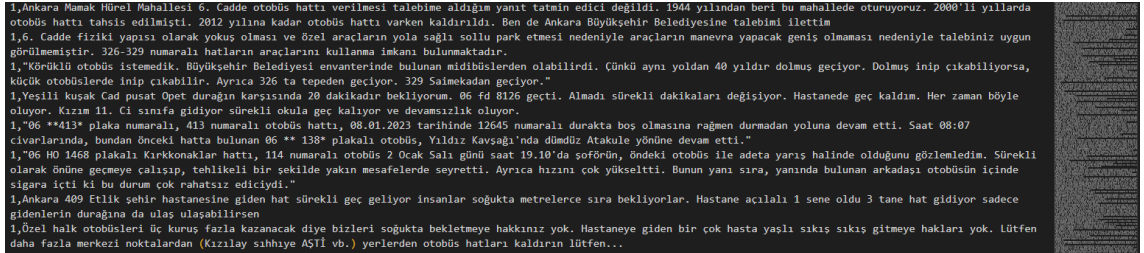
```
from datasets import load_dataset  
  
from transformers import AutoTokenizer, AutoModelForSequenceClassification  
from transformers import TrainingArguments, Trainer  
  
import numpy as np  
import evaluate
```

Şekil 3.2. Kütüphanelerin Yüklenmesi

3.3. Veri Seti

Modelin eğitim ve doğrulama sürecinde kullanılan veriler manuel olarak, açık kaynaklı web sitelerinden elde edilmiştir. Veriler, farklı şehirlerdeki kişiler tarafından farklı belediyelere yazılan gerçek şikayet metinlerinden oluşmaktadır. Toplanan bu veriler 8 farklı sınıf etiketi ile manuel olarak etiketlenmiştir. Veri setinin model eğitiminde kullanılabilmesi için CSV dosya formatı tercih edilmiştir. Toplam veri miktarı, eğitim (train) veri seti için 2698 adet ve doğrulama (validation) veri seti için 289 adet olarak belirlenmiştir. Uygulama kapsamında belirlenmiş sınıflar şu şekildedir:

- Su ve Kanalizasyon
- Ulaşım, Park ve Bahçeler
- Zabıta
- Temizlik İşleri
- Veteriner İşleri
- Sosyal Hizmet ve Yardımlar
- Diğer



1,Ankara Mamak Mülki Mahallesi 6. Cadde otobüs hattı verilmesi talebime aldığım yanıt tatmin edici değildi. 1944 yılından beri bu mahallede oturuyoruz. 2000'li yıllarda otobüs hattı tahsis edilmişti. 2012 yılına kadar otobüs hattı varken kaldırıldı. Ben de Ankara Büyükşehir Belediyesine talebimi ilettim

1,6. Cadde fiziki yapısı olarak yokuş olması ve özel araçların yola sağlı sollu park etmesi nedeniyle araçların manevra yapacak geniş olmaması nedeniyle talebiniz uygun görülmemiştir. 326-329 numaralı hatların araçlarını kullanma imkanı bulunmaktadır.

1,"Körükli otobüs istemedik. Büyükşehir Belediyesi envanterinde bulunan midibüslerden olabildi. Çünkü aynı yoldan 40 yıldır dolmuş geçiyor. Dolmuş inip çıkabiliyorsa, küçük otobüslerde inip çıkabilir. Ayrıca 326 ta tepeden geçiyor. 329 Saimkadan geçiyor."

1,Yesilli kuşak Cad pusat Opet durağın karşısında 20 dakikadır bekliyorum. 06 fd 8126 geçti. Almadı sürekli dakikaları değişiyor. Hastanede geç kaldım. Her zaman böyle oluyor. Kızım 11. Cı sınıfa gidiyor sürekli okula geç kalıyor ve devamsızlık oluyor.

1,"06 **413* plaka numaralı, 413 numaralı otobüs hattı, 08.01.2023 tarihinde 12645 numaralı durakta boş olmasına rağmen durmadan yoluna devam etti. Saat 08:07 civarlarında, bundan önceki hatta bulunan 06 ** 138* plakalı otobüs, Yıldız Kavşağı'nda dümdüz Atakule yönüne devam etti."

1,"06 HD 1468 plakalı Kırkkonaklar hattı, 114 numaralı otobüs 2 Ocak Salı günü saat 19.10'da şoförün, öndeki otobüs ile adeta yarış halinde olduğunu gözlemledim. Sürekli olarak önüne geçmeye çalışıp, tehlikeli bir şekilde yakın mesafelerde seyretti. Ayrıca hızını çok yükseltti. Bunun yanı sıra, yanında bulunan arkadaşı otobüsün içinde sigara içti ki bu durum çok rahatsız ediciydi."

1,Ankara 409 Etilik şehir hastanesine giden hat sürekli geç geliyor insanlar soğukta metrelerce sıra bekliyorlar. Hastane açılalı 1 sene oldu 3 tane hat gidiyor sadece gidenlerin duruşuna da ulaşabiliyrsen

1,Özel halk otobüsleri üç kuruş fazla kazanacak diye bizleri soğukta bekletmeye hakkınız yok. Hastaneye giden bir çok hasta yaşlı sıkış sıkış gitmeye hakları yok. Lütfen daha fazla merkezi noktalardan (Kızılay sıhhiye AŞİ vb.) yerlerden otobüs hatları kaldırın lütfen...

Şekil 3.3. Veri Setinden Veri Örnekleri

Hazırlanan eğitim (train) ve doğrulama (validation) veri setlerinin modelin eğitiminde kullanılabilmesi için proje koduna dahil edilmesi gerekmektedir. Bu işlem Python'ın datasets kütüphanesinin "load_dataset()" fonksiyonu kullanılarak gerçekleştirilmiştir.

```
train_dataset = load_dataset("csv", data_files={"train": "train.csv"})
val_dataset = load_dataset("csv", data_files={"validation": "validation.csv"})
```

Şekil 3.4. Veri Setlerinin Yükleme

3.4. Pretrained Model Ve Tokenizer'ın Yüklmesi

Modelin eğitimi için bir pretrained model ve bu modelin sağlamış olduđu tokenizer Hugging Face Hub'dan yüklenmiştir. Kullanılan bu model, büyük miktarda Türkçe veriler ile eğitilmiş BERT tabanlı bir modeldir.

```
tokenizer= AutoTokenizer.from_pretrained("dbmdz/bert-base-turkish-uncased")  
model= AutoModelForSequenceClassification.from_pretrained("dbmdz/bert-base-turkish-uncased", num_labels=8)
```

Şekil 3.5. Pretrained Model Ve Tokenizer

3.5. Veri Setlerinin Model Eğitime Hazırlanması

CSV formatında hazırlanmış veriler model eğitiminde ham hallerinde kullanılamazlar. Bu metin verilerinin tokenlara dönüştürülmesi gerekmektedir. Bu işlem bir tokenizer fonksiyonu ile gerçekleştirilmektedir. Tokenizer fonksiyonları metni daha küçük birimlere ayırmak için kullanılır; Bu birimler; kelimeler, alt kelimeler, ekler ve kökler biçiminde olabilir. Bu işlem, metinlerin yapısal analizini kolaylaştırır. Tokenlara dönüştürme işlemi yapılırken farklı uzunluktaki metin verilerinin modelde kullanılması gerekebilir. Böyle bir durumda bu metinler sabit bir uzunluğa göre ayarlanmalıdır. Kısa olan metin verilerindeki boşluklar padding tokenları ile doldurulmalıdır. Bu işlemleri gerçekleştirmek için bir önceki bölümde pretrained model ile birlikte yüklenmiş olan tokenizer fonksiyonu kullanılmıştır. Ayrıca modelin genelleme performansının artırılması amacı ile veri setindeki veriler rastgele bir sırada karıştırılmıştır. Bu işlemlerin sonucunda veri setleri modelin eğitimi için hazır hale getirilmiştir.

```
def tokenize_function(examples):  
    return tokenizer(examples["text"], padding="max_length", truncation=True)  
  
tokenized_train_dataset = train_dataset.map(tokenize_function, batched=True)  
tokenized_validation_dataset = val_dataset.map(tokenize_function, batched=True)  
  
tuned_train_dataset = tokenized_train_dataset["train"].shuffle(seed=42)  
tuned_eval_dataset = tokenized_validation_dataset["validation"].shuffle(seed=42)
```

Şekil 3.6. Veri Setlerinin Ön İşlenmesi

3.6. Değerlendirme Metriği

Modelin eğitimi sırasında ve eğitimin sonucunda validation veri seti üzerinde sınıf tahminlerini ne kadar doğru yaptığını takip edebilmek ve model performansını değerlendirebilmek amacı ile bir değerlendirme metriğine ihtiyaç duyulmaktadır. Bu nedenle proje kapsamında bir doğruluk metriği tanımlanmıştır. Bu metrik kullanılarak doğruluk oranı ve kayıp değeri gibi performans ölçütlerini hesaplayabilmek için `compute_metrics` fonksiyonu tanımlanmıştır.

```
metric = evaluate.load("accuracy")

def compute_metrics(eval_pred):
    logits, labels = eval_pred
    predictions = np.argmax(logits, axis=-1)
    return metric.compute(predictions=predictions, references=labels)
```

Şekil 3.7. Değerlendirme Metriği

3.7. Eğitim Parametrelerinin Ayarlanması

Parametre değerleri model eğitim sürecinin önemli bir bileşenidir. Eğitim sürecini ve sonuçlarını etkileyecek birçok farklı parametre ayarı yapılabilir. Proje kapsamında transformers kütüphanesine ait “TrainingArguments” sınıfı kullanılarak bazı parametre değerleri belirlenmiştir.

```
training_args = TrainingArguments(
    output_dir="test_trainer",
    evaluation_strategy="epoch",
    num_train_epochs=3,
    learning_rate=0.0001
)
```

Şekil 3.8. Parametre Değerleri

3.8. Eğitim

Proje kapsamında bir eğitim döngüsü oluşturmak için transformers kütüphanesine ait “Trainer” sınıfı ve “train()” fonksiyonu kullanılmıştır. Train sınıfının parametre değerlerine

model, parametreler, veri setleri ve değerlendirme metriği gibi daha önceden hazırlanan bileşenler verilmiştir. Ardından train fonksiyonu ile eğitim döngüsü başlatılmıştır. Eğitim süreci bittiğinde modelin kayıt edilip kullanılabilmesi için “save_pretrained()” fonksiyonu kullanılmıştır. Eğitim süreci 3 epoch olarak belirlenmiştir. Tüm eğitim süreci 4 GB ekran kartına sahip kişisel bilgisayarda gerçekleştirilmiştir. Bu süreç yaklaşık 17 saat almıştır.

```
trainer = Trainer(  
    model=model,  
    args=training_args,  
    train_dataset=tuned_train_dataset,  
    eval_dataset=tuned_eval_dataset,  
    compute_metrics=compute_metrics,  
)  
  
trainer.train()  
fine_tuned_model = trainer.model  
fine_tuned_model.save_pretrained("./modelPack")
```

Şekil 3.9. Eğitim Döngüsü

3.9. Arayüz

Eğitilmiş ve kayıt edilmiş olan modelin gerçek hayattaki kullanımına örnek teşkil etmesi açısından localhost'ta çalışan bir web arayüzü tasarlanmıştır. Bu amaç doğrultusunda python'ın flask kütüphanesi kullanılmıştır. Arka planda çalışan model, yazılan şikayet metnini sınıflandırmaktadır. Sonuç ekranında sınıflandırma sonucunu ve ilgili birime ait iletişim numarasını dinamik bir şekilde kullanıcıya sunmaktadır.



Akdeniz Belediyesi

Şikayetlerinizi Bize İletin..

Şikayetinizi Bize İletin.

Çocuk parklarının içler acısı durumu tüm mahalleliyi rahatsız ediyor. Her yerde çekirdek kabukları, ambalaj atıkları var. Görevlilerden ilgilenilmesini rica ediyoruz.

Gönder

Şekil 3.10. Şikayet Arayüzü



Akdeniz Belediyesi

Şikayetinizin Temizlik İşleri birimimiz ile ilgili olduğunu düşünmekteyiz.

Bir sorun olduğunu düşünüyorsanız
aşağıdaki butondan şikayet sayfasına geri
dönerek yeni bir şikayet metni
oluşturabilirsiniz.

Size daha iyi hizmet verebilmemiz
için lütfen
Temizlik İşleri
birimimiz ile iletişime geçiniz.

Temizlik İşleri
Birimi Dahili Numarası:
0324 336 55 55

Başka Bir Şikayet İletin

Şekil 3.11. Sonuç Arayüzü

4. SONUÇLAR

Bu tez kapsamında gerçekleştirilmiş ve incelenmiş olan sınıflandırma modeli, belediyeçilik alanında karşılaşılan istek ve şikayetlerin yönetimi konusunda bir çözüm ortaya koymuştur. İstek ve şikayetlerin ilgili birimler ile daha doğru ve hızlı bir şekilde ilişkilendirilmesini sağlayıp, hizmet sürelerini önemli ölçüde kısaltabileceği konusunda fikirler vermiştir. İlerleyen dönemlerde bu konu üzerinde geliştirmeler yapılarak belediyeçilikte kullanılabilecek bir şikayet yönetim sistemi haline getirilebilmesi mümkündür. Hem vatandaşlar hem de kurumlar açısından böyle bir sistemin gelecekte gerçekçi faydaları olabilir.

Bu tez, vatandaşlar ve belediyeler arasında etkili bir iletişimin geliştirilmesinde yapay zekanın nasıl katkıları olabileceğini incelemiştir. Bu amaç doğrultusunda gerçekleştirilmiş olan metin sınıflandırma modeli, **%91.4** doğruluk oranı ve **0.16** kayıp değeri sonuçlarını vermiştir. Kayıp değeri kabul edilebilir seviyelerde olsa da, olması gereken ideal değerden daha yüksek olduğu için modelin genelleştirme yeteneğinin geliştirmeye açık olduğu sonucu ortaya çıkmaktadır.

Ön eğitilmiş bir model üzerinde fine-tuning yönteminin, gerek bu tezde incelendiği gibi sınıflandırma problemlerinde, gerekse doğal dil işlemenin kullanıldığı diğer problemlerde çözüm geliştirmek için oldukça kullanışlı bir yöntem olduğu anlaşılmıştır. Kelimelerin bağlamsal anlamlarının daha iyi anlaşılması açısından Transformer Mimarisi ve BERT modelinin kullanılmış olması, projenin başarısına büyük katkılar sağlamıştır.

Bu tez kapsamında gerçekleştirilen modelin gelişmeye açık tarafları da bulunmaktadır. Daha büyük ve daha çeşitli konulardaki şikayet metinlerinden derlenmiş bir veri seti ile çalışmak başarılı tahmin ve genelleştirme yeteneğini artırabilir. Bir belediye ile işbirliği yapılarak vatandaşın gelen şikayetlerin tutulduğu veri tabanlarına erişim sağlanarak ilgili belediye özelinde daha doğru sonuçlar veren ve daha optimize bir sistemin hayata geçirilmesi mümkündür.

5. KAYNAKLAR

- [1] Alpaydın, E. (2013). Yapay Öğrenme. İstanbul: Boğaziçi Üniversitesi Yayınevi.
- [2] Mccarthy, J. (2004). What Is Artificial Intelligence?. Erişim Adresi (11 Ocak 2019): [Http://Wwwformal.Stanford.Edu/Jmc/Whatisai/](http://Wwwformal.Stanford.Edu/Jmc/Whatisai/).
- [3] Nabiye, V. V. (2012). Yapay Zeka: İnsan-Bilgisayar Etkileşimi. Ankara: Seçkin Yayıncılık.
- [4] Nilsson, N. (1990). The Mathematical Foundations Of Learning Machines. San Mateo: Morgan Kaufmann.
- [5] E. Öztemel, “Yapay Sinir Ağları,” Papatyayayincilik, İstanbul, 2003.
- [6] S. Murat, B. Mühendisliği, A. Dalı, And Y. Lisans, “İnsansız Hava Aracı Görüntülerinden Derin Öğrenme Yöntemleriyle Nesne Tanıma Yüksek Lisans Tezi,” Maltepe Üniversitesi, Lisansüstü Eğitim Enstitüsü, 2021. Accessed: May 02, 2021. [Online]. Available: [Http://Openaccess.Maltepe.Edu.Tr/Xmlui/Handle/20.500.12415/7379](http://Openaccess.Maltepe.Edu.Tr/Xmlui/Handle/20.500.12415/7379).
- [7] F. Doğan And İ. Türkoğlu, “Derin Öğrenme Modelleri Ve Uygulama Alanlarına İlişkin Bir Derleme,” Dümf Mühendislik Derg., Vol. 10, No. 2, Pp. 409–445, Jun. 2019, Doi: 10.24012/Dumf.411130.
- [8] Y. Lecun, Y. Bengio, And G. Hinton, “Deep Learning,” Nature, Vol. 521, No. 7553. Nature Publishing Group, Pp. 436– 444, May 27, 2015, Doi: 10.1038/Nature14539.
- [9] R. Elshawi, A. Wahab, A. Barnawi, And S. Sakr, “Dlbench: A Comprehensive Experimental Evaluation Of Deep Learning Frameworks,” Clust. Comput. J. Networks, Softw. Tools Appl., P. 1, 2021, Doi: 10.1007/S10586-021-03240-4.
- [10] F. Chollet, Deep Learning With Phyton. 2018
- [11] S. Albawi, T. A. Mohammed, And S. Al-Zawi, “Understanding Of A Convolutional Neural Network,” In Proceedings Of 2017 International Conference On Engineering And Technology, Icet 2017, Mar. 2018, Vol. 2018-Janua, Pp. 1–6, Doi: 10.1109/Icengtechnol.2017.8308186.
- [12] X. Wu, D. Sahoo, And S. C. H. Hoi, “Recent Advances In Deep Learning For Object Detection,” Neurocomputing, Vol. 396, Pp. 39–64, 2020, Doi: 10.1016/J.Neucom.2020.01.085.
- [13] C. Min, J. Xu, L. Xiao, D. Zhao, Y. Nie, And B. Dai, “Attentional Graph Neural Network For Parking-Slot Detection,” Ieee Robot. Autom. Lett., Vol. 6, No. 2, Pp. 3445–3450, Apr. 2021, Doi: 10.1109/Lra.2021.3064270.
- [14] Ö. İnik And E. Ülker, “Derin Öğrenme Ve Görüntü Analizinde Kullanılan Derin Öğrenme Modelleri,” Gaziosmanpaşa Bilim. Araştırma Derg., Vol. Issn, No. 6.3, 2017.
- [15] Malkoç, B. (2012). Temel Bilimler Ve Mühendislik Eğitiminde Programlama Dili Olarak Python. Xiv. Akademik Bilişim Konferansı Bildirileri, 201.
- [16] Ong, Charlene J., Agni O., Rebecca Z., Francois Pierre M. C., Meghan H., Liang M., Darian F., Vd., 2020, “Machine Learning Ve Natural Language Processing Methods To Identify Ischemic Stroke, Acuity Ve Location From Radiology Reports.”, Plos One, 15 (6):
- [17] Zhou, M., Nan D., Shujie L., Ve Heung-Yeung S., 2020, “Progress In Neural Nlp: Modeling, Learning, Ve Reasoning.”, Engineering, 6 (3): 275–90. [Https://Doi.Org/10.1016/J.Eng.2019.12.014](https://doi.org/10.1016/J.Eng.2019.12.014).

- [18] Joachims, T., “A Probabilistic Analysis Of The Rocchio Algorithm With Tfidf For Text Categorization”, Proceedings Of The Fourteenth International Conference On Machine Learning, San Francisco, Ca, Usa, 143–151, 1997.
- [19] Oğuzlar, A., ”Metin Madenciliği Nedir?”, Temel Metin Madenciliği, Bursa, Dora Basım, 2011.
- [20] Haltaş, A., Alkan, A., & Karabulut, M. (2015). Metin Sınıflandırmada Sezgisel Arama Algoritmalarının Performans Analizi. Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi, 30(3), 417-427.
- [21] Y. K., Akın, Veri Madenciliğinde Kümeleme Algoritmaları Ve Kümeleme Analizi, Doktora Tezi, Marmara Üniversitesi, Sosyal Bilimler Enstitüsü, 2008.
- [22] İ. F., Pilavcılar, Metin Madenciliği İle Metin Sınıflandırma, Yüksek Lisans Tezi, Yıldız Teknik Üniversitesi, Fbe Matematik Mühendisliği Anabilim Dalı, 2007.
- [23] G., Guo, H., Wang, D., Bell, Y., Bi, K., Greer, “Using Knn Model For Automatic Text Categorization”, Soft Computing, 10 (5), 423-430, 2006.
- [24] Z., Deng, M., Zhang, “Improving Text Categorization Using The Importance Of Words In Different Categories”, Computational Intelligence And Security, Xi'an, Çin, 458-463, 15-19 Aralık, 2005.
- [25] S., Manne, S. K., Kotha, And S.S., Fatima, “Text Categorization With K-Nearest Neighbor Approach”, Information Systems Design And Intelligent Applications 2012, Visakhapatnam, Hindistan, 413-420, Ocak 2012.
- [26] H. K., Yıldız, M., Gençtav, N., Usta, B., Diri, M. F., Amasyalı, “Metin Sınıflandırmada Yeni Özellik Çıkarımı”, 15. Sinyal İşleme, İletişim Ve Uygulamaları Kurultayı, Eskişehir, 11-13 Haziran, 2007.
- [27] V. C., Gandhi, J. A., Prajapati, “Review On Comparison Between Text Classification Algorithms”, International Journal Of Emerging Trends & Technology In Computer Science (Ijettcs), 1 (3), 2012.
- [28] S. M., Weiss, C., Apte, F. J., Damerau, D. E., Johnson, F. J., Oles, T., Goetz, T., Hampp, “Maximizing Text-Mining Performance” Ieee Intelligent Systems And Their Applications, New York, 14(4), 63- 69, 1999.
- [29] K., Wu, B. L., Lu, M., Utiyama, H., Isahara, “An Empirical Comparison Of Min–Max-Modular K-Nn With Different Voting Methods To Large-Scale Text Categorization”, Soft Computing, 12(7), 647-655, 2008.
- [30] A., Kehagias, V., Petridis, V.G., Kaburlasosand, P., Fragkou, “A Comparison Of Word And Sense-Based Text Categorization Using Several Classification Algorithms”, Journal Of Intelligent Information Systems, 21(3), 227-247, 2001.
- [31] İnternet: S., Albayrak, Veri Madenciliği Sınıflama Ve Kümeleme Yöntemi, [Http://Www.Ce.Yildiz.Edu.Tr/Personal/Songul/File/332/Veri+Madencili%C4%9fi-S%C4%B1n%C4%B1flamakumeleme.Ppt](http://www.ce.yildiz.edu.tr/Personal/Songul/File/332/Veri+Madencili%C4%9fi-S%C4%B1n%C4%B1flamakumeleme.Ppt), 17.03.2013.
- [32] [Https://Huggingface.Co/Blog/Bert-101](https://Huggingface.Co/Blog/Bert-101)
- [33] [Https://Huggingface.Co/Learn/Nlp-Course/Chapter1/4#How-Do-Transformers-Work](https://Huggingface.Co/Learn/Nlp-Course/Chapter1/4#How-Do-Transformers-Work)
- [34] [Https://Huggingface.Co/Learn/Nlp-Course/Chapter1/4#Transfer-Learning](https://Huggingface.Co/Learn/Nlp-Course/Chapter1/4#Transfer-Learning)

6. ÖZGEÇMİŞ

Yasin GÜNAY

01.03.2000 Türkiye/MERSİN Doğumlu

Fırat Üniversitesinde Mühendislik Fakültesi/Bilgisayar Mühendisliği bölümünde 4. sınıf öğrencisi.

Python, PHP, laravel, PostgreSQL teknolojilerinde deneyimli.

B1 Seviye İngilizce yabancı dil bilgisine sahip.

İletişim Bilgileri:

Telefon: +90 553 443 71 89

Email: ygunay055@gmail.com

Açık Adres: Nailbey mahallesi, Soylu sokak, 10/1, kat: 3 daire:4 MERKEZ/ELAZIĞ