T.C.
DOKUZ EYLÜL ÜNİVERSİTESİ
SOSYAL BİLİMLER ENSTİTÜSÜ
YÖNETİM BİLİŞİM SİSTEMLERİ ANABİLİM DALI
YÖNETİM BİLİŞİM SİSTEMLERİ PROGRAMI
YÜKSEK LİSANS TEZİ

ORGANİZASYONEL GELİŞİM İÇİN OTOMATİK KONUŞMA TANIMA SİSTEMİ ÖNERİSİ

Davut Emre TAŞAR

Danışman
Dr. Öğr. Üyesi Kutan KORUYAN

İZMİR – 2023

TEZ ONAY SAYFASI

YEMİN METNİ

Yüksek Lisans Tezi olarak sunduğum “Organizasyonel Gelişim İçin Otomatik Konuşma Tanıma Sistemi Önerisi” adlı çalışmanın, tarafımdan, akademik kurallara ve etik değerlere uygun olarak yazıldığını ve yararlandığım eserlerin kaynakçada gösterilenlerden oluştuğunu, bunlara atıf yapılarak yararlanılmış olduğunu belirtir ve bunu onurumla doğrularım.

…./…../2023

Davut Emre TAŞAR

ÖZET

Yüksek Lisans Tezi

Organizasyonel Gelişim İçin Otomatik Konuşma Tanıma Sistemi Önerisi

Davut Emre TAŞAR

Dokuz Eylül Üniversitesi

Sosyal Bilimler Enstitüsü

Yönetim Bilişim Sistemleri Anabilim Dalı

Yönetim Bilişim Sistemleri Programı

Otomatik konuşma tanıma (ASR) teknolojisinin kullanımı son yıllarda hızla artmıştır ve birçok kuruluş çeşitli iş fonksiyonlarında verimliliği, doğruluğu ve üretkenliği artırmak için onu benimsemektedir. Bu tez çalışması, ASR sistemlerinin organizasyonel faydalarını keşfetmekle birlikte, organizasyonların rekabet avantajı elde etmek için bu faydalardan nasıl yararlanabileceğine dair bir model önerisi ve önerilen model ile birlikte bir çağrı merkezindeki performans artışını ölçümleme önerisi sunmayı amaçlamaktadır.

ASR sistemleri, konuşulan dili metne dönüştürmek için gelişmiş algoritmalar ve doğal dil işleme teknikleri kullanarak kuruluşların veri girişini, transkripsiyonu ve konuşma dilini içeren diğer görevleri otomatikleştirmesini sağlar. Bu görevlerin doğruluğunu ve verimliliğini artırabilir, manuel redaksiyon ve transkripsiyon ihtiyacını azaltabilir ve zamandan ve kaynaklardan tasarruf sağlayabilir.

Doğal dil işleme, bilişim sistemlerini kullanarak, hedeflenen dil veya diller ile ilgili metin ve ses verilerinin matematiksel temsillerini oluşturarak, bu temsiller üzerinden çeşitli algoirtmalar vasıtası ile dil verisini bilgiye, bilgiyi de bilgeliğe dönüştürme yöntemlerine verilen genel bir kavramdır. COVID-19 pandemisi sonrasında kullanım sıklığında artış meydana gelen uzaktan çalışma ve eğitim modelleri içerisinde sıkça yapılan online toplantılar nedeni ile ASR sistemlerinin önemi ve bu sistemlerin uygulama sayısı artış göstermektedir. Ancak, ASR sistemlerinin yapay zekâ algoritmaları ile eğitimi söz konusu olduğunda, üretilen modellerin başarımı, kullanılan yapay zekâ algoritmaları

kadar, eğitim verilerinin yeterliliğine bağlanabilir. ASR sistemleri, ses dosyalarını ve karşılık gelen doğru şekilde yazılmış transkript dosyalarını içeren büyük miktarda açıklamalı eğitim verisi gerektirir.

Bu açıklamalı (etiketli) eğitim verilerinin çoğu dil için doğrulanmış bir şekilde bulunması zordur, genellikle insanların ses doyalarının karşılıklarını girmelerini manuel olarak gerçekleştirmesini gerektirir, bu da maliyetinin dışında hataya da açıktır. Bu senaryo için denetimli bir eğitim görevi pratik değildir. Türkçe, bol miktarda etiketlenmiş veriye sahip olmayan dillerden biridir ve bu da ASR sisteminin doğruluğunu İngilizce, Fransızca veya İspanyolca gibi kaynak açısından zengin diğer dillere kıyasla çok düşük kılar. Bu çalışmada, kendi kendini denetleyen bir görevde veya eğitim öncesi aşamada etiketlenmemiş eğitim verilerinden (yalnızca ses dosyaları) genel veri temsillerini öğrenerek etiketlenmemiş ses verilerinden yararlanılmaktadır. Bu aşama, girdiyi maskeleyen ve karşıt bir görevi çözen Wav2Vec 2.0 mimarisi kullanılarak gerçekleştirilmiştir. Metin karşılıkları bulunan Türkçe verileri kullanarak Türkçe dilinde ince ayar yapmak amacıyla Wav2Vec 2.0 kullanarak farklı dillerde önceden eğitilmiş modellerden de yararlanılmaktadır. Birden çok dilde ham konuşma dalga biçimleri üzerinde, Türkçe veriler üzerinde eğitilmiş birçok dilli modele ince ayar yapılarak 0,23 oranında düşük bir kelime hata (WER) oranı elde edilmiştir. Böylelikle elde edilen bu konuşma tanıma modeli, organizasyonların süreçlerini yönetim bilişim sistemleri kullanarak geliştirmeyi hedefleyen bir ürün ortaya çıkartmış ve bunu açık kaynaklı olarak paylaşılmıştır.

Anahtar Kelimeler: Wav2Vec2, Otomatik Konuşma Tanıma, Konuşmadan Metne Dönüşüm, Doğal Dil İşleme, Organizasyonel Gelişim

ABSTRACT

Master's Thesis

An Automatic Speech Recognition System Proposal for Organizational Development

Davut Emre TAŞAR

Dokuz Eylül University

Graduate School of Social Sciences

Department of Management Information Systems

Management Information Systems Program

The use of automatic speech recognition (ASR) technology has grown rapidly in recent years, and many organizations are adopting it to increase efficiency, accuracy and productivity in various business functions. While exploring the organizational benefits of ASR systems, this thesis aims to propose a model on how organizations can take advantage of these benefits to gain competitive advantage, and to measure performance improvement in a call center with the proposed model.

ASR systems use advanced algorithms and natural language processing techniques to convert spoken language to text, enabling organizations to automate data entry, transcription, and other tasks involving spoken language. This can improve the accuracy and efficiency of tasks, reduce the need for manual proofreading and transcription, and save time and resources.

Natural language processing is a general concept given to the methods of transforming language data into knowledge and knowledge into wisdom by using information systems, creating mathematical representations of text and audio data related to the targeted language or languages, through various algorithms over these representations. The importance of ASR systems and the number of applications of these systems increase due to the frequent online meetings within the distance working and education models, which have increased in the frequency of use after the COVID-19 pandemic. However, when it comes to training ASR systems with artificial intelligence algorithms, the performance of

the produced models can be attributed to the adequacy of the training data as well as the artificial intelligence algorithms used. ASR systems require large amounts of annotated training data, which includes audio files and corresponding correctly written transcript files.

This annotated (tagged) training data is difficult to find in a validated form for most languages, often requiring people to manually enter equivalents of audio files, which is not only costly but also error-prone. A supervised training task is impractical for this scenario. Turkish is one of the languages that does not have abundant labeled data, which makes the accuracy of the ASR system very low compared to other resource-rich languages such as English, French or Spanish. This study utilizes unlabeled audio data by learning general data representations from unlabeled training data (audio files only) in a self-monitoring task or pre-training phase. This phase is accomplished using the Wav2Vec 2.0 architecture, which masks the input and solves an opposite task. Pre-trained models in different languages are also utilized using Wav2Vec 2.0 to fine-tune the Turkish language using Turkish data with text equivalents. On the raw speech waveforms in multiple languages, a low word error (WER) rate of 0.23 was achieved by fine-tuning the multilingual model trained on the Turkish data. This speech recognition model thus obtained has created a product that aims to improve the processes of organizations by using management information systems, and it has been shared as open source.

Keywords: Wav2Vec2, Automatic Speech Recognition, Speech-to-text, Natural Language Processing, Organizational Development

ORGANİZASYONEL GELİŞİM İÇİN OTOMATİK KONUŞMA TANIMA SİSTEMİ ÖNERİSİ

İÇİNDEKİLER

TEZ ONAY SAYFASI ii
YEMİN METNİ iii
ÖZET iv
ABSTRACT vi
İÇİNDEKİLER viii
KISALTMALAR x
TABLOLAR LİSTESİ xii
ŞEKİLLER LİSTESİ xiii

GİRİŞ 1

BİRİNCİ BÖLÜM
LİTERATÜR TARAMASI

1.1. MAKİNE ÖĞRENMESİ 5
1.2. YAPAY SİNİR AĞLARI 6
1.3. TEKRARLAYAN SİNİR AĞLARI 7
1.4. EVRİŞİMLİ SİNİR AĞLARI 10
1.5. DÖNÜŞTÜRÜCÜ 13
1.6. OTOMATİK KONUŞMA TANIMA 16
1.6.1. Geleneksel Otomatik Konuşma Tanıma 16
1.6.2. Uçtan Uca Otomatik Konuşma Tanıma 18
1.6.3. Çağrı Merkezlerinde Otomatik Konuşma Tanıma Kullanılması 19
1.6.4. Organizasyonel Gelişimde Otomatik Konuşma Tanıma Sistemlerinin Rolü 20

İKİNCİ BÖLÜM
VERİ, YÖNTEM VE KISITLAR

2.1. VERİ SETİ 21
2.2. YÖNTEM 23
2.2.1. Kendinden Denetimli Akustik Modelleme 23
2.2.1.1. Wav2Vec 2.0 24
2.2.1.2. Kelime Hata Oranı 26
2.2.2. Sistem Geliştirme Yaşam Döngüsü 26
2.2.2.1. Problemlerin Tanımlanması 27
2.2.2.2. Bilgi Gereksinimlerin Belirlenmesi 28
2.2.2.3. Sistem İhtiyaç Analizi 28
2.2.2.4. Sistem Tasarımı 29
2.2.2.5. Yazılımın Geliştirilmesi 32
2.2.2.6. Test 32
2.3. KISITLAR 32

ÜÇÜNCÜ BÖLÜM
UYGULAMANIN GERÇEKLENMESİ

3.1. UYGULAMA YAZILIMININ GERÇEKLENMESİ 33
3.2. VERİ, JETONLAR, ÖZNİTELİK SETLERİNİN OLUŞTURULMASI 33
3.3. MODELİN EĞİTİMİ 38
3.4. MODELİN AÇIK KAYNAKLI PAYLAŞIMI 40

SONUÇ 41
KAYNAKÇA 44

KISALTMALAR

ASR	Otomatik Konuşma Tanıma (Automatic Speech Recognition)
WER	Kelime Hata Oranı (Word Error Rate)
L2	İkinci Dil (Language two)
LM	Dil Modeli (Language Model)
AI	Yapay Zekâ (Artificial Intelligence)
CV	Bilgisayarlı Görme (Computer Vision)
GPT	Türetici Ön Eğitimli Dönüştürücü (Generative Pre-trained Transformer)
NLP	Doğal Dil İşleme (Natural Language Processing)
E2E	Uçtan Uca (End to end)
ML	Makine Öğrenmesi (Machine Learning)
DL	Derin Öğrenme (Deep Learning)
YSA	Yapay Sinir Ağları (Artifical Neural Networks)
DNN	Derin Sinir Ağları (Deep Neural Networks)
FFNN	İleri Beslemeli Sinir Ağları (Feed Forward Neural Networks)
MLP	Çok Katmanlı Tahminleyici (Multi-layer Perceptron)
SLP	Tek Katmanlı Tahminleyici (Single-layer Perceptron)
RNN	Tekrarlayan Sinir Ağları (Recurrent Neural Networks)
LSTM	Uzun Kısa Süreli Bellek (Long Short Term Memory)
CNN	Evrişimli Sinir Ağları (Convolutional Neural Networks)
ReLU	Düzeltilmiş Doğrusal Birim (Rectified Linear Unit)
EOS	Cümle sonu (End of Sentence)
MLM	Maskelenmiş Dil Modeli (Masked Language Modelling)
MFCC	Mel Frekans Cepstral Katsayıları (Mel-frequency cepstrum)
HPF	Yüksek Geçiş Filtresi (Hyper-pass Filter)
DFT	Ayrık Fourier Dönüşümü (Diverse Fourier Transformer)
DCT	Ayrık Kosinüs Dönüşümü (Diverse Cosine Transformer)
HMM	Saklı Markov Modelleri (Hidden Markov Models)
GMM	Gauss Karışım Modelleri (Gauissian Mixture Models)
TDNN	Zaman Gecikmeli Sinir Ağları (Time Delayed Neural Networks)

SAMPA	Konuşma Değerlendirme Yöntemleri Fonetik Alfabesi (The Speech Assessment Methods Phonetic Alphabet)
CTC	Bağlamsal Zamansal Sınıflandırma (Connectionist Temporal Classification)
NMT	Sinirsel Makine Çevirisi (Neural Machine Translation)
SSAM	Kendinden Denetimli Akustik Modelleme (Self Supervised Acuistic Modelling)
VQ-VAE	Vektör Nicemli Varyasyonel Otomaik Kodlayıcı (Vector-Quantized Variational Autoencoders)
GELU	Gauss Linear Hata Ünitesi (The Gaussian Error Linear Unit)
SGYD	Sistem Geliştirme Yaşam Döngüsü
CER	Harf Hata Oranı (Character Error Rate)
SSL	Kendinden Denetimli Öğrenme (Self Supervised Learning)
XLSR	Çapraz Dilli Konuşma Temsilleri (Cross-lingual Speech Representations)
UUTK	Uçtan uca otomatik konuşma tanıma
GKKT	Geleneksel otomatik konuşma tanıma
WER	Kelime Hata Oranı (Word Error Rate)
GPU	Grafik İşlem Birimi

TABLOLAR LİSTESİ

Tablo 1: Veri Setinin Analizi s.22

Tablo 2: Modelin Performans Sonuçları s.39

ŞEKİLLER LİSTESİ

Şekil 1: MLP Mimarisinin Bir Gösterimi s.7
Şekil 2: Standart RNN Mimarisinin Bir Gösterimi s.8
Şekil 3: RNN Biriminin ve LSTM Biriminin İç Yapısının Bir Gösterimi s.9
Şekil 4: LeNet-5 CNN Modeli Mimarisi s.11
Şekil 5: Dönüştürücü Mimarisinin Bir Gösterimi s.14
Şekil 6: Geleneksel ASR Boru Hattı s.17
Şekil 7: Wav2Vec 2.0 Çerçevesinin Çizimi s.25
Şekil 8: Sistem Geliştirme Yaşam Döngüsü s.27
Şekil 9: Sistem Yazılım Mimarisi - 1 s.30
Şekil 10: Sistem Yazılım Mimarisi - 2 s.31
Şekil 11: Veri Setinin İndirilmesi s.33
Şekil 12: Kullanılmayan verilerin Ayıklanması ve Veri Setinden Örnekler s.34
Şekil 13: Ön İşleme Tabi Tutulmuş Transkripsyon Verileri s.35
Şekil 14: Öznitelik Dönüştürücü Fonksiyon s.36
Şekil 15: Orijinal ve Dönüşmüş Ses Dosyaları ve Transkripsyonları s.37
Şekil 16: Modelin performans skorları s.39
Şekil 17: Yazar Tarafından Paylaşılan Model s.40

GİRİŞ

ASR, bilgisayarların konuşulan dili anlamasını ve yorumlamasını sağlayan bir teknolojidir. İnsan-makine etkileşimini sağlamayı amaçlayan ASR, onlarca yıldır yoğun bir araştırma alanı olmuştur. ASR, konuşma tanımanın çeşitli algoritmalar sayesinde otomatik bir şekilde gerçekleşebilmesini mümkün kılar ve bu sayede sinyal verisinden bilgi üretebilme becerisini elde etmeyi sağlar. Elde edilen bu bilgi organizasyonel gelişim için çok önemli bir veri kaynağı niteliği teşkil eder. Bu konudaki teknolojik gelişmeler ilk olarak, Gauss karışım modelleri, gizli Markov modelleri, mel-frekans kepstral katsayıları ve türevleri, n-gram dil modelleri gibi birçok temel teknolojinin geliştirilmesi ile başlamıştır (Yu ve Deng, 2016).

Bu teknikler, ASR'de ve ilgili alanlarda son teknolojiyi büyük ölçüde geliştirmiştir. Daha önceki başarılarla karşılaştırıldığında, 2010'dan önceki yıllarda ASR'nin araştırılması ve uygulanmasındaki ilerleme yavaşken, ancak bu dönemde pratik sistemlerde gizli Markov modelleri, mel frekans kepstral katsayıları ve türevleri dizisi ayırt edici eğitimi gibi önemli tekniklerin optimize edilerek ASR problemleri çözülmeye çalışmıştır. Son birkaç yılda, mobil cihazlarda ASR'ye olan talebin artması ve mobil dünyada sesli arama, mesaj dikte ve sanal konuşma asistanları (örn. Apple'ın Siri'si, Google Now ve Microsoft'un Cortana'sı) sayesinde ASR konusunun popularitesi artmış, konu ile ilgili yapılan çalışmalar hız kazanmıştır (Yu ve Deng, 2016).

ASR kullanmanın gelişmiş verimlilik, artan üretkenlik ve gelişmiş müşteri deneyimleri dahil olmak üzere birçok faydası vardır (Zekveld ve diğerleri, 2009: 262).

ASR, toplantıları yazıya dökmek veya müşteri sorularını yanıtlamak gibi görevleri otomatikleştirerek kuruluşların zamandan tasarruf etmesine ve personelini daha önemli görevlere odaklanması için serbest kalmasını sağlar. Bu, organizasyonun üretkenliği ve verimliliğinin artmasına yol açan bir gelişim yönüdür. (Danis ve Karat, 1995: 17)

ASR'nin bir başka yararı da kuruluşların müşteri deneyimlerini geliştirmelerine yardımcı olabilmesidir. ASR, müşterilerin doğal dil işleme algoritmalarını da kullanarak kuruluşlarla etkileşim kurmasını sağlayıp, müşterilerin ihtiyaç duydukları bilgileri almasını, soru sormasını ve talepte bulunmasını daha kolay ve daha verimli

hale getirebilir. Bu, müşteri memnuniyetinin ve bağlılığının artmasına neden olur. (Danis ve Karat, 1995: 17)

ASR ayrıca, müşteri davranışı ve tercihlerine ilişkin değerli içgörüler sağlayarak kuruluşların rekabet avantajı elde etmelerine yardımcı olabilir. ASR, müşterilerin konuşma dilini analiz ederek kuruluşların müşterilerinin ne aradığını, endişelerinin ne olduğunu ve onlara nasıl daha iyi hizmet verebileceklerini anlamalarına yardımcı olabilir. Bu, kuruluşların ürün ve hizmetlerini müşterilerinin ihtiyaçlarını daha iyi karşılayacak şekilde uyarlamalarına yardımcı olabilir.

Yukarıda belirtilen faydalara ek olarak, ASR kullanmanın diğer bazı organizasyonel faydaları da vardır. Bunlar şunları içerir:

- **Geliştirilmiş doğruluk ve güvenilirlik:** Kuruluşların hataları azaltmasına ve verilerinin doğruluğunun artırmasına yardımcı olabilir. ASR, konuşulan dili otomatik olarak yazılı metne çevirerek kuruluşların verilerinin doğru ve güncel olmasını sağlamalarına yardımcı olabilir.
- **Gelişmiş güvenlik:** Ses biyometrisi ve kimlik doğrulama gibi gelişmiş güvenlik özellikleri sağlamak için kullanılabilir. ASR, kuruluşların müşterilerinin ve çalışanlarının kimliklerini benzersiz ses kalıplarını kullanarak doğrulamasını sağlayarak, kuruluşların dolandırıcılığı önlemesine ve hassas verileri korumasına yardımcı olabilir (Kabir ve diğerleri, 2021: 79236).
- **Artırılmış erişilebilirlik:** İşitme engelli bireyler için gerçek zamanlı transkripsiyon ve metinden konuşmaya sentezi gibi erişilebilirlik özellikleri sağlamak için kullanılabilir. ASR, bu kişilerin konuşulan dili kullanarak kuruluşlara erişmesini ve onlarla etkileşim kurmasını sağlayarak kuruluşların tüm müşterileri için daha iyi hizmet ve destek sağlamasına yardımcı olabilir.

Genel olarak, ASR, her büyüklükteki kuruluşun etkinlik, üretkenlik ve müşteri deneyimlerini geliştirmesine yardımcı olabilir. Sıkıcı ve zaman alan görevleri otomatikleştirerek, veri doğruluğunu iyileştirerek ve gelişmiş güvenlik ve erişilebilirlik sağlayarak kuruluşların hedeflerine ulaşmalarına ve günümüzün rekabetçi iş ortamında başarılı olmalarına yardımcı olabilir.

Son yıllarda, ASR teknolojisindeki performans artışı neredeyse insan çevirmenlerin seviyesine ulaşmıştır. (Amodei ve diğerleri, 2016: 173; Xiong ve diğerleri, 2018: 5934; Chiu ve diğerleri, 2018: 4775). Yapılan araştırmalar, kendi

kendini denetleyen ön eğitim tekniklerinin akustik verilere de uygulanabileceğini göstermiştir (Van Den Oord ve Vinyals, 2017: 30; Zhao ve diğerleri, 2021: 1). Bu çalışmalarda, kendini denetleyen ön eğitim tekniklerinin gösterimleriyle, kayda değer miktarda özgün konuşma verisi kullanılarak eğitilmiş ve daha sonra, örneğin konuşma tanıma gibi bir alt görev için kullanılmıştır. Bu sayede uçtan uca ASR sistemleri, birçok dilde veya tek bir dil özelinde büyük bir külliyat kullanılarak, Wav2Vec2.0 (Baevski ve diğerleri, 2020: 12449) mimarisi ile açık kaynaklı olarak oluşturulup, hedeflenen dil üzerinde ince ayar (fine-tuning) yapılarak, dil özelinde modellerin oluşturulması sağlanmıştır.

Literatürde Türkçe diline yönelik yapılan ASR çalışmaları ise sınırlıdır. Koruyan'ın çalışmasında canlı yayınlanan internet yayınları için Google'ın Web Speech to Text API kullanılarak otomatik altyazı oluşturulması için bir öneri sunulmuştur. Çalışmada, konuşanın mikrofona olan uzaklığı, ortam gürültüsü, hızlı veya kesintisiz konuşma kullanılan modelin performansına olan etkileri irdelenmiştir (Koruyan, 2015: 1).

Yakar ve Aşlıyan çalışmalarında, Saklı Markov Modeli kullanılarak Türkçe dili için konuşma tanıma modelleri eğitilmiş ve performansları analiz edilmiştir. WOE (kelime hata oranı) ile analiz edilen modelde test için kullanılan kısıtlı bir veri seti ile %17,4 oranında bir başarı elde edilmiştir. Ayrıca modelin performansını arttırmak için modelin tahminlediği heceleri birleştirerek, Türkçe sözcük arama işlemi gerçekleştiren art-işleme (post-processing) algoritması kullanılmıştır (Yakar ve Aşlıyan, 2016: 1).

Oyucu, Saadin, Polat ve Sever, çalışmalarında, Hirsch historgramı farkı ile konuşmalardaki gürültülerin tespiti için kullanılmış, bu amaçla bir eşik değer belirlenerek, örneklem frekansından da faydalanarak gürültü ve sessizlik tespiti yapılmıştır. Bu şekilde gürültü ve sessizliklerden arındırılan veri seti, C++ dilinde yazılmış olan kaldi kütüphanesi (Povey, 2011; 1) kullanılarak bir konuşma tanıma sistemi oluşturulmuştur. Bu yaklaşımla eğitilen konuşma tanıma modelinin, ön işlemeye tabi tutulmadan eğitilen modele göre kelime hata oranının %7,41 oranında arttığı görülmüştür (Oyucu ve diğerleri, 2020: 334).

Türkçe dili özelinde yapılan çalışmalar incelendiğinde, farklı dillerdeki otomatik konuşma tanıma alanında yukarıdaki çalışmalar incelendiğinde klasik yöntemler kullanılarak sonuca ulaşılmaya çalışılmıştır. Fakat, Wav2Vec2 mimarisinin

diğer dillerdeki yüksek başarısı göz önünde bulundurulduğunda ve ayrıca, literatürde Türkçe'ye yönelik Wav2Vec2 mimarisi kullanılarak yapılan bir çalışma bulunmadığından, çalışma sonuçlarının analizinin literatüre faydalı olacağı kanaati bu bu tez çalışmasının çıkış noktasını oluşturmuştur.

Bu tez çalışması ile dönüştürücü tabanlı 53 dile ait konuşma verileri kullanılan ön eğitimli Wav2Vec2 modeli, öğrenme aktarması yöntemi ile açık kaynaklı bir veri seti olan, Mozilla Common Voice veri setinin Türkçe diline ait verileri kullanılarak ve ince ayar işlemine tabi tutularak Türkçe özelinde çalışabilen bir konuşmadan metne dönüştürücü modeli geliştirilmiş, geliştirilen uygulamanın performans analizleri ve kısıtları tespit edilip, geliştirilen uygulamanın olası kullanım alanları üzerinde durulmuştur. Daha sonra ise geliştirilen bu modelin, çağrı merkezi işleten bir organizasyonda verimliliği arttırmada kullanmak üzere yazılım mimarisi tasarımı yapılmıştır.

BİRİNCİ BÖLÜM
LİTERATÜR TARAMASI

Doğal dil işleme kapsamında konuşmanın metne dönüştürülmesi konusu araştırmacılar tarafından ilgi çeken ve gelişmekte olan bir alandır. Literatür incelediğinde benzer amaçları olan ve farklı yöntemleri kullanan çalışmalar ile karşılaşılmıştır. Bu çalışmalar farklı dillere ve alanlara katkı sağlamışlardır. Son yıllarda yapay zekâ alanındaki gelişmelerin hızlanması ile konuşma tanıma sistemlerinde sözcük/hece tabanlı geleneksel yöntemlerden ziyade öğrenme tabanlı yaklaşımlarda bir artış görülmektedir. Konuyu temelinden ele alarak, ilgili kavramlar genelden özele doğru ilerleyecek bir şekilde literatür taraması gerçekleştirilmiştir. Literatür taramasının sonunda ise Türkçe dilinde yapılan çalışmalardan ve otomatik konuşma tanıma sistemi ile çağrı merkezlerinde yapılan uygulamalardan bahsedilmiştir.

1.1. MAKİNE ÖĞRENMESİ

Makine öğrenimi, sistemlerin deneyimlerinden otomatik olarak öğrenmesini ve geliştirmesini sağlayan bir yapay zekâ alanıdır. Bu algoritmalar, minimum insan müdahalesi ile eğitim verileri olarak da adlandırılan önemli miktarda bilgiden kalıpları öğrenmeyi amaçlar. Eğitim sürecinde bu kalıplara göre, makine öğrenimi sistemleri daha önce görülmemiş yeni veriler üzerinde tahminler ve kararlar verebilir. Böylece sistemler kendilerini deneyimlerine dayalı olarak otomatik olarak geliştirir ve günceller.

Derin öğrenme, YSA'na dayalı algoritmalara odaklanan bir makine öğrenimi alt alanıdır. Makine öğrenimi algoritmaları tipik olarak her zaman optimal olmayan el yapımı özellikler gerektirir. Derin öğrenme sistemleri ise özellikleri otomatik olarak çıkarır ve hatalarını öğrenerek daha az insan müdahalesi gerektirir. Bununla birlikte, karmaşık yapı ve çok sayıda öğrenilebilir parametre nedeniyle, derin öğrenme algoritmaları geleneksel makine öğrenme sistemlerine göre daha fazla eğitim verisi gerektirir.

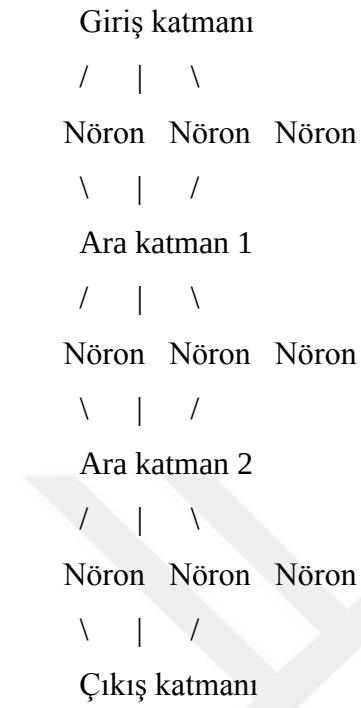
1.2. YAPAY SİNİR AĞLARI

YSA, insan beyninin ve sinir sisteminin çalışma prensiplerini taklit eden bir model ailesidir. Bu modeller, Rosenblatt (1958) tarafından önerilen bir algılayıcı kavramını kullanarak biyolojik sinir sistemlerini yapay olarak taklit eder. YSA'ların giriş ve çıkış katmanları arasında birden fazla katman bulunabilir (Rosenblatt, 1958: 386). Bu durumda bu tip ağlara DNN adı verilir. Bir algılayıcı, sinir ağının bir katmanını oluşturmak için bir yapı taşıdır ve karşılık gelen ağırlıklar ve bir sapma, ağırlıklı bir toplam ve bir etkinleştirme işlevi ile tek veya çoklu girdilerden oluşan bir hesaplama birimidir. Önyargı (Bias) olarak tanımlanan ve sabit bir rakam olarak hesaplama birimi içerisinde bulunan b, algılayıcının çıktıyı üretmeden önce ulaşması gereken bir eşik değeridir. Önyargı, bir etkinleştirme işlevine geçmeden önce girişlerin ağırlıklı toplamına eklenir. Etkinleştirme işlevi ise çıktının genliğini belirli bir aralıkta sınırlayan matematiksel bir işlevdir (McCulloch ve Pitts, 1943: 117).

FFNN'ler, yapay sinir ağlarının ilk ve en basit türüdür. Bu ağlarda, bilgi herhangi bir döngü olmaksızın yalnızca ileri yönde akar. FFNN'nin en basit sınıfı, bir giriş ve bir çıkış katmanından oluşan tek katmanlı bir algılayıcıdır. Katmanlar tamamen bağlantılıdır, yani bir katmandaki her bir düğüm, bir sonraki katmandaki her bir düğümle bağlantılıdır.

FFNN'lerin başka bir sınıfı, bir giriş ve bir çıkış katmanının yanı sıra bir veya daha fazla gizli katmandan oluşan çok katmanlı bir algılayıcıdır. Ara katmanlar, ağ girişlerinden veya çıkışlarından doğrudan gözlemlenemedikleri için gizli olarak adlandırılır. Tek katmanlı tahminleyici SLP'lerde olduğu gibi, MLP'lerde katmanlar tamamen birbirine bağlıdır. Bir MLP'nin genel mimarisi Şekil 1'de gösterilmektedir.

Şekil 1: MLP Mimarisinin Bir Gösterimi



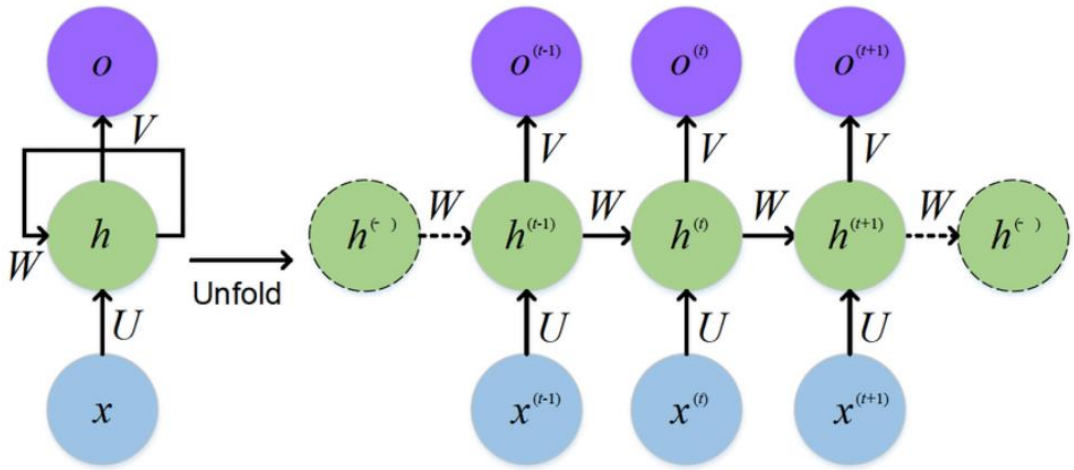
Kaynak: (Yazar Tarafından Derlenmiştir).

1.3. TEKRARLAYAN SİNİR AĞLARI

RNN sıralı verileri işlemek için tasarlandı. Bu ağlarda, herhangi bir döngü yoktur ve bu nedenle, mevcut zaman adımındaki çıktı, önceki zaman adımlarındaki girdilere bağlı olmaz. Bu ağlarda, mevcut katmanın çıktısını hesaplamak için daha önceki girdilerden elde edilen hesaplamaları geri besleyerek bir dahili bellek kavramı uygulanır. RNN'lerde, giriş dizisinin geçmiş bilgileri gizli durumda saklanır ve güncellenir.

Bir RNN'nin iç mimarisi Şekil 2'de gösterilmiştir. Bu ağların iç mimarisi, girdilerin tek tek işlendiği ve çıktıların birbirine bağlı olarak hesaplandığı düz bir şekildedir. Ağın her bir zaman adımında bir veya daha fazla nöron bulunur ve her bir nöron, girdileri ağırlıklı olarak toplayarak bir çıktı üretir. Bu çıktı, bir sonraki zaman adımında kullanılır ve ağdaki dahili belleği günceller. Bu sayede, ağ, sıralı verilerin geçmiş bilgilerini hesaba katarak daha anlamlı çıktılar üretebilir.

Şekil 2: Standart RNN Mimarisinin Bir Gösterimi

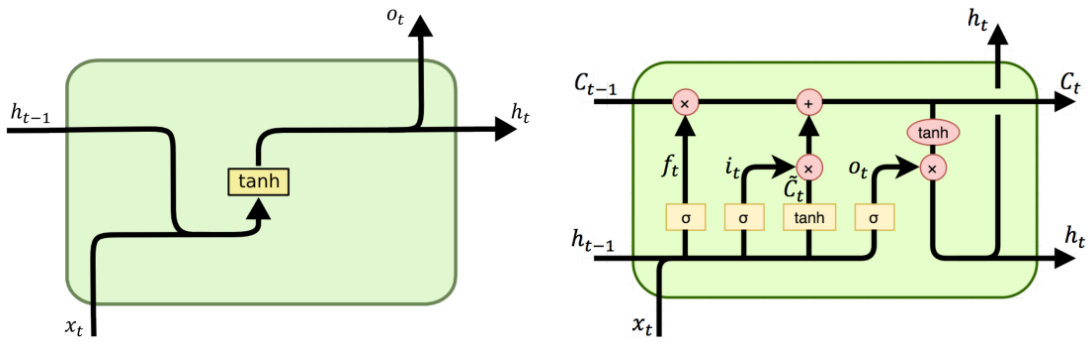


Kaynak: (Feng ve diğerleri, 2017: 682).

RNN, patlama olarak tanımalanan çok yüksek değerlere ulaşan veya kaybolan diye tanımlanan çok düşük değerlere ulaşan gradyanlardan muzdariptir. Bu problemler, aşırı derecede büyük gradyan değerlerinin öğrenme sürecini engellediği ve gradyanın geri yayılım sırasında küçüldüğü ve öğrenmeye katkıda bulunmayı bıraktığı durumlardır. Bu sorunların çözümü için LSTM ağları önerilir (Hochreiter ve Schmidhuber, 1997: 1736). LSTM, doğrusal olmayan kapılara sahip bir geçit mekanizması kullanarak ezberleme sürecini kontrol eden standart bir RNN'nin bir uzantısıdır. LSTM ağları, RNN'lere kıyasla genişletilmiş belleğe sahiptir ve uzun vadeli bağımlılıkları işleyebilir.

Şekil 3, standart bir RNN biriminin iç yapısını bir LSTM birimindekiyle karşılaştırır. Standart bir LSTM birimi, bir unutma kapısı, bir giriş kapısı ve bir çıkış kapısından oluşur.

Şekil 3: Standart Bir RNN Biriminin (Solda) ve Bir LSTM Biriminin (Sağda) İç Yapısının Bir Gösterimi



(a) Tekrarlayan Sinir Ağları Birimi (b) LSTM – Uzun Kısa Süreli Bellek Birimi

Kaynak: (Olah, 2015).

Unutma kapısı, hangi bilgilerin atılabileceğini keşfeder. Geçerli girdi vektörü x_t'yi ve önceki gizli durum h_{t-1}'i sigmoid etkinleştirme işlevinden geçirir ve vektör f_t'yi 0'dan 1'e kadar bir ölçekte değerler verir, burada 0 "tamamen unut" ve 1 "tamamen tut" anlamına gelir.

Giriş kapısı, yeni bilgilerin hangi bölümlerinin önemli olduğuna ve belleği değiştirmek için kullanılması gerektiğine karar verir. Geçerli girdi x_t ve önceki gizli durum h_{t-1} üzerinde sigmoid uygular ve girdinin daha önemli kısımlarının daha büyük değerler aldığı çıktıyı verir. Giriş kapısının çıkışı daha sonra x_t ve h_{t-1}'in tanh aktivasyon fonksiyonuna beslenmesi ile elde edilen hücre giriş aktivasyon vektörü ile eleman bazında çarpılır.

Çıkış kapısı, mevcut gizli durumu h_t hesaplar. Güncellenen hücre durumu c_t, gizli durumda saklanması gerekmeyen bilgileri içerdiğinden, çıkış kapısı, c_t'nin hangi bölümlerinin h_t içinde korunması gerektiğini ve hangi bölümlerin atılabileceğini keşfeder. Çıkış kapısı ilk önce mevcut girdi x_t'yi ve önceki gizli durumu h_{t-1}'i işler ve o_t üretir. Gizli durum h_t, yeni değiştirilmiş hücre durumu C_t'nin o_t ve tanh'ın Hadamard ürünüdür.

1.4. EVRİŞİMLİ SİNİR AĞLARI

CNN, yapay sinir ağlarının bir alt türüdür ve görüntü işleme gibi görsel verileri işlemek için tasarlanmıştır (Matsumoto ve diğerleri, 1990: 100). CNN, doğal dil işleme, ses işleme veya genel veri işleme gibi diğer uygulamalarda da kullanılabilir (Yao ve diğerleri, 2019: 7370; Abdel-Hamid ve diğerleri, 2014: 1533).

Bununla birlikte, bu tür mimari, girdinin matris formuna uygun bir şekilde dönüştürülmesinden sonra, örneğin NLP görevleri (Kalchbrenner ve diğerleri, 2014: 2188) gibi diğer derin öğrenme uygulamalarında da kullanılır. CNN'ler ilk kez 1979'da Neocognitron (Fukushima, 1980: 193) adlı bir sinir ağında tanıtıldı.

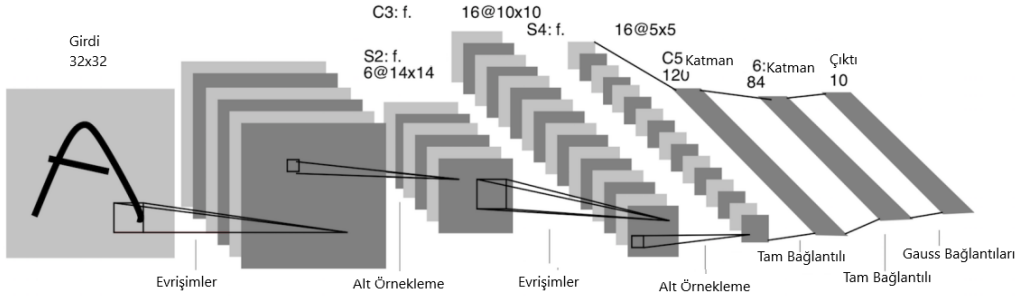
CNN, sinir ağlarına benzer bir yapıya sahiptir, ancak ağın gizli katmanlarında özel bir tasarım kullanılır (Wu, 2017: 5). Bu tasarım, doğal görüntülerin yapısını ve özelliklerini daha iyi öğrenmeye yardımcı olur. Örneğin, görüntüler genellikle yüksek derecede özelleşmiş ve lokal olarak birbirinden farklı özellikler içerir. Bu özellikler, farklı bölgelerde farklı boyutlarda ve farklı düzenlerde olabilir. CNN bu özellikleri öğrenmek için lokal bağlantıları ve paylaşımlı ağırlıkları kullanır. Bu, ağın, girdi görüntüsündeki benzer özellikleri öğrenmesine izin verir ve ağın eğitimine daha az veri gerektirir.

CNN genellikle birkaç gizli katman, birçok filtre ve bir düşürme işlemi kullanır. Filtreler, girdi görüntüsündeki özellikleri bulmak için kullanılır ve düşürme işlemi, özelliklerin boyutunu azaltarak ağın performansını ve hızını artırır. Bu işlemler, ağın girdi görüntüsünden çıktı görüntüsüne dönüştürülmesini sağlar ve bu çıktı görüntüsü, ağın girdi görüntüsündeki özellikleri tanımak için kullanılabilir (Wu, 2017: 495).

Şekil 4, el yazısı ve rakam sınıflandırması için 1998'de tanıtılan basit bir CNN modeli LeNet-5'i göstermektedir. LeNet-5, birincil girdisi görüntü ve ikincil girdisi de etiketleri içerir. Öğrenme sürecinde, ağ, girdi görüntüsünü tarar ve görüntünün özelliklerini öğrenir. Öğrenme sürecini tamamladıktan sonra, ağ, girdi görüntüsünün etiketini tahmin edebilir. LeNet-5, birden fazla gizli katman, birçok filtre ve bir düşürme işlemi kullanır ve bu işlemler, ağın girdi görüntüsünden çıktı görüntüsüne dönüştürülmesini sağlar. LeNet-5 ağı, genellikle giriş katmanı, iki konvolüsyonel katman, iki azaltma katmanı ve bir çıkış katmanından oluşur. Giriş katmanı, girdi

görüntüsünü alır ve bu görüntüyü küçük parçalara böler. Konvolüsyonel katmanlar ise, giriş katmanından gelen parçaları tarar ve bu parçalarda bulunan özellikleri öğrenir. Azaltma katmanları ise, girdi görüntüsündeki özelliklerin yoğunluğunu azaltarak, ağın öğrenme sürecini hızlandırır. Son olarak, çıkış katmanı, ağın öğrendiği özellikleri kullanarak girdi görüntüsünün etiketini tahmin eder.

Şekil 4: LeNet-5 CNN Modeli Mimarisi



Kaynak: (LeCun ve diğerleri, 1998:2278)

LeNet-5 ağının matematiksel ifadesi aşağıdaki gibidir:

- Giriş katmanı: X, Konvolüsyonel katmanlar: C1, C2, Azaltma katmanları: S2, S4
- Çıkış katmanı: F6, Giriş katmanından gelen girdi görüntüsü X, birinci konvolüsyonel katman C1 tarafından taranır ve bu katmanın çıktısı C1 = f(X * W1 + b1) ile ifade edilebilir.

Burada, W1 ve b1 ağırlıkları ve sapma değerlerini, f ise bir etkinleştirme işlevini temsil eder. İkinci konvolüsyonel katman C2 ise, birinci katmanın çıktısından gelen verileri tarar ve bu katmanın çıktısı C2 = f(C1 * W2 + b2) ile ifade edilebilir:

İkinci konvolüsyonel katmanın çıktısı, birinci azaltma katmanı S2 tarafından işlenir ve bu katmanın çıktısı S2 = g(C2) ile ifade edilebilir. Burada, g bir azaltma işlevini temsil eder. İkinci azaltma katmanı S4 ise, ikinci konvolüsyonel katmanın çıktısından gelen verileri işler ve bu kat manın çıktısı S4 = g(C2) ile ifade edilebilir.

Dördüncü azaltma katmanı S4'ün çıktısı, çıkış katmanı F6 tarafından işlenir ve bu katmanın çıktısı F6 = h(S4 * W3 + b3) ile ifade edilebilir. Burada, W3 ve b3 ağırlıkları ve sapma değerlerini, h ise bir etkinleştirme işlevini temsil eder. Bu şekilde, LeNet-5 ağı, girdi görüntüsünün özelliklerini öğrenir ve bu özellikleri kullanarak girdi görüntüsünün etiketini tahmin eder.

Evrişim doğrusal bir dönüşüm olduğu için, ağa doğrusal olmayanlığı dahil etmek için genellikle evrişim katmanında bazı etkinleştirme işlevleri uygulanır. Goodfellow 'a göre, ReLU $f(x) = \max(0, x)$, CNN gibi modern sinir ağları için varsayılan öneridir. Bununla birlikte, örneğin tanh veya sigmoid gibi başka aktivasyon fonksiyonları da uygulanabilir (Goodfellow ve diğerleri, 2016: 330).

Havuzlama (Pooling) katmanı, CNN'in bir parçası olarak, girdi tensörünün boyutlarını azaltarak önemli özellikleri korur. Genellikle, max havuzlama veya ortalama havuzlama olarak adlandırılan iki temel yöntem kullanılır. Max havuzlama, bir bellek kümesinin her bir alt dizisi için en büyük değeri seçerken, ortalama havuzlama, bellek kümesinin her bir alt dizisindeki elemanların ortalamasını alır. Bu yöntemler, girdi tensörünün boyutlarını azaltarak, önemli özellikleri korur ve böylece ağın çıktısının doğruluğunu artırır (Goodfellow ve diğerleri, 2016: 333).

Evrişim katmanı dışında, Havuzlama katmanında öğrenilebilir parametre yoktur. Bunun yerine, özellik haritalarının bölümlerine uygulanacak önceden tanımlanmış bir havuzlama işlemi vardır. Havuzlama işlemindeki en popüler işlevler, bölümlerin her birinden en büyük öğeyi seçen maksimum havuzlama ve tensörün her bir parçası için ortalama değeri hesaplayan ortalama havuzlamadır.

Ağın son evrişim veya havuzlama katmanının çıktısı genellikle düzleştirilir veya tek boyutlu bir vektöre yeniden şekillendirilir ve bir veya daha fazla tam bağlantılı katmana iletilir. Tam bağlantılı katmanların amacı, özellik haritalarını, örneğin, son katmanın çıktı boyutunun sınıf sayısına karşılık geldiği sınıflandırma gibi, nihai çıktıya eşlemektir. Tam bağlantılı katmanlar, normal bir ileri beslemeli sinir ağında olduğu gibi, bir katmanın her bir nöronunu bir sonraki katmanın her bir nöronuna bağlar. Bu katmanların her birini, örneğin ReLU gibi bir etkinleştirme işlevi takip eder. Son tam bağlı katmana uygulanan doğrusal olmayan işlev, ağın son görevine bağlıdır. Örneğin, çok sınıflı sınıflandırmada Softmax $f(x_i) = \exp(x_i) / \sum_j \exp(x_j)$ Yamashita'ya göre aktivasyon fonksiyonu için tipik bir seçimdir, çünkü çıkış değerlerinin normalizasyonunu gerçekleştirir. Son tam bağlantılı katman, çıktıları 1'e kadar negatif olmayan değerlere dönüştürerek, böylece onları her bir hedef sınıfın olasılıklarına dönüştürme işlemini gerçekleştirir (Yamashita ve diğerleri, 2018: 611).

1.5. DÖNÜŞTÜRÜCÜ

Dönüştürücü (Transformer) yapısı, ilk olarak "Attention is all you need" adlı bir makalede tanıtılmıştır (Vaswani ve diğerleri, 2017: 5998). Bu makalede, Dönüştürücü yapısı, düğümler arasındaki bağlantıların önem derecelerini hesaplamak için öz dikkat (self-attention) mekanizmasını kullanan ilk yapı olarak tanımlanmıştır. Daha sonra, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding" adlı bir makalede, Dönüştürücü yapısının kullanımı, derin bidireksiyonel dönüştürücü ön-eğitimleri ile dil anlama görevlerinde kullanılmıştır (Devlin ve diğerleri, 2018: 1).

Dönüştürücü yapısının mimarisi, ilk olarak girdi verilerinin şifreleyici(encoder) ve çözücü(decoder) kısımlara ayrılması ile başlar. Şifreleyici kısmında, girdi verileri vektör nokta çarpımları da denen öz dikkat(self-attention) mekanizması kullanılarak, düğümler arasındaki bağlantıların önem derecelerini belirleyen bir çıktı oluşturulur. Bu çıktı, çözücü kısmında, bir çıktı verisi oluşturmak için kullanılır (Vaswani ve diğerleri, 2017: 5998).

Dönüştürücü yapısında, şifreleyici ve çözücü kısımları da aynı yapıya sahiptir. Her bir şifreleyici ve çözücü kısmı, birden fazla "katlama" bloğundan oluşur. Her bir katlama bloğu, self-attention mekanizmasını içeren birçok başlı dikkat bloğu ve bir pozisyon bazlı ileri beslemeli nöron bloğundan oluşur.

Çok başlı dikkat bloğu, öz-dikkat mekanizmasını içerir ve girdi verilerindeki düğümler arasındaki bağlantıların önem derecelerini hesaplar. FFN bloğu ise, MHA bloğundan gelen çıktıyı işler ve girdi verilerinin düğümleri arasındaki bağlantıların önem derecelerini kullanarak, çıktı verisi oluşturur.

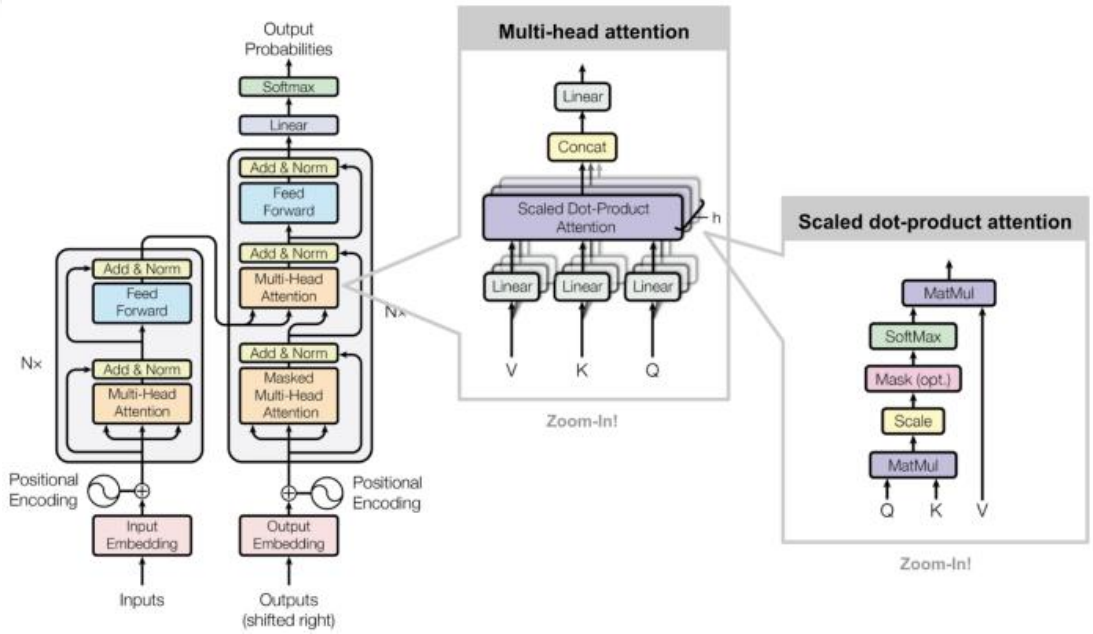
Dönüştürücü yapısı, bu şekilde, öz-dikkat mekanizmasını kullanarak girdi verilerini işler ve bu sayede, düğümler arasındaki bağlantıların önem derecelerini hesaplar.

Dönüştürücülerde, girdi verilerini öz-dikkat mekanizması ile işleyerek, düğümler arasındaki bağlantıların önem derecelerini hesaplar. Bu mekanizma sayesinde, Dönüştürücü, RNN'lerle karşılaştırıldığında daha hızlı bir şekilde NLP görevlerini gerçekleştirir. Bu yapının mimarisi, şifreleyici ve çözücü kısımlarından

oluşur ve her bir şifreleyici ve çözücü kısmı da birden fazla "katlama" bloğundan oluşur. Her bir katlama bloğu ise, MHA ve FFN bloğlarından oluşur.

Dönüştürücü mimarisi Şekil 5'te gösterilmiştir. Kodlayıcının amacı, $x = (x_1, x_2, ..., x_n)$ giriş verilerini, farklı segmentlere seçici olarak katılarak $z = (z_1, z_2, ..., z_n)$ yüksek boyutlu kodlamalara eşlemektir. Giriş dizilerinde, kod çözücünün görevi $z = (z_1, z_2, ..., z_n)$ kodlayıcı temsillerini $y = (y_1, y_2, ..., y_m)$ çıkış dizilerine dönüştürmektir. Hem kodlayıcı hem de kod çözücü, aynı yapılara sahip ancak paylaşılmayan ağırlıklara sahip birden çok katman yığınından oluşur.

Şekil 5: Dönüştürücü Mimarisinin Gösterimi. (Çok Başlı Dikkat ve Ölçeklenmiş Nokta-Ürün Dikkati Dahil)



Kaynak: (Vaswani, 2017: 6000)

Kodlayıcı katmanı iki bölümden oluşur: Kendine dikkat eden bir alt katman (bkz. Şekil 5'in orta kısmı) ve tam bağlantılı bir ileri beslemeli ağ. Her alt katmanın artık bir bağlantısı (F(x) + x) vardır ve bunu katman normalizasyonu takip eder (Ba ve diğerleri, 2016: 1). Öz-dikkat, Dönüştürücü'nün belirli bir belirteci kodlarken giriş sırasındaki diğer belirteçlere katılmak için uyguladığı tekniktir.

Şekil 5'in sağ tarafında tek bir dikkat katmanı veya kafa gösterilmektedir. Şekilde Q, K ve V sırasıyla sorgu, anahtar ve değer matrislerini temsil etmektedir.

Bunlar, öz-dikkat katmanına dahil edilen ağırlık matrisleriyle girdi gömme matrisinin çarpılmasıyla hesaplanır: $W^Q \in R^{dmodel \times dk}$, $W^K \in R^{dmodel \times dk}$ ve $W^V \in R^{dmodel \times dv}$.

Q, K ve V verildiğinde, kendine dikkat katmanının çıktısı şu şekilde hesaplanır: Attention (Q, K, V) = softmax ($QK^T / d_k^{0.5}$)* V

Q, K ve V elde edildikten sonra, tüm anahtarlarla sorgunun nokta ürünleri hesaplanır, d_k'ya bölünür ve softmax işleviyle normalleştirilir. Gradyanları daha kararlı hale getirmek için nokta çarpımlar ($1 / d_k^{\wedge 0.5}$) faktörü ile ölçeklendirilir. Bu ölçekleme faktörü, softmax işlevinin aşırı derecede küçük gradyanlara sahip olduğu alanlara girmesini engeller. Elde edilen dikkat ağırlıkları (softmax çıktısı), matris çarpımı ile V değer matrisinde kodlanan dizideki tüm tokenlara uygulanır.

Dönüştürücü, tek bir dikkat başlığı uygulamak yerine, çok başlı dikkat mekanizmasını kullanır. Öz-dikkat katmanı h = 8 paralel dikkat başlığı içerir. Her birinin, girdileri farklı temsil altuzaylarına yansıtan kendi sorgu, anahtar ve değer ağırlık matrisleri vardır. Birden fazla dikkat katmanından etkilenen ekstra hesaplama maliyetini azaltmak için sorguların, anahtarların ve değerlerin boyutu azaltılarak tekrar hesaplanır ve $d_k = d_v = d_{model} / h = 64$'e düşürülür.

Kod çözücü bloğu, etraflarında artık bağlantılar ve üstlerinde katman normalizasyonu bulunan üç alt katmandan oluşur. Kodlayıcı tarafında kullanılan bir öz-dikkat ve ileri besleme alt katmanına ek olarak, giriş dizisinin farklı bölümleriyle ilgilenen başka birçok başlı dikkat öğesi vardır. Bu bileşen, önceki katmandan kendi sorgu matrisini Q hesaplaması dışında, öz-dikkat alt katmanına benzer şekilde çalışır, ancak anahtar ve değer dikkat matrisleri K ve V, üst kodlayıcı çıkışından alınır. Kod çözücüdeki öz-dikkat alt katmanı da değiştirilir: çıktı dizisinin gelecekteki konumlarına katılmasını kısıtlamak için softmax adımından önce maskeleme ile desteklenir.

Kod çözücü yığınının üstünde, kod çözücü çıktısını model sözlüğünün her bir simgesi için puanları olan bir vektör olarak kabul edilebilecek logit vektörüne dönüştüren doğrusal bir katman vardır. Softmax katmanı daha sonra puanları olasılıklara dönüştürür ve en yüksek olasılığa sahip token seçilir. Bu belirteç, kod çözücü giriş dizisine eklenir ve bir sonraki çıktıyı üretmek için kod çözücüye iletilir. Kod çözme işlemi, kod çözücü özel, cümle sonu (EOS) belirtecini verene kadar yinelemeli olarak tekrarlanır.

Çözücü girişi de bir konum sağa kaydırılır. Aslında, birinci konum, bu nedenle kod çözücüye iletilecek olan ilk belirteç olan bir cümle başlangıcı belirteci ile doldurulur. Bu nedenle, hedef sekans, kaydırma olmaksızın (başlangıçta cümle başı belirteci yok) ve sonunda bir EOS belirteci olan kod çözücü girişidir.

Kaynak dizileri kodlayıcıya ve hedef dizileri kod çözücüye beslemeden önce, bunlar düşük boyutlu vektör temsillerine veya gömmelere dönüştürülür. Standart dönüştürücü, önceden eğitilmiş yerleştirmeleri kullanmak yerine öğrenilmiş girdi ve çıktı yerleştirme katmanlarını uygular. Ayrıca girdi ve çıktı gömme katmanları arasında ağırlık matrisini paylaşır. Bu katmanlar belirteçleri d_{model} = 512 boyutlu vektörlere dönüştürür (Vaswani ve diğerleri, 2017: 6000).

Dönüştürücü herhangi bir yineleme veya evrişim içermediğinden, jetonlar hakkındaki konum bilgisini kodlamak için konumsal gömmeler eklenir. Bu kodlamalar, sırasıyla kodlayıcıya ve kod çözücü yığınlarına iletilmeden önce girdi ve çıktı gömmeleriyle toplanır. Modeldeki belirteçlerin sırasını enjekte etmek için sinüsoidal fonksiyonlar kullanılır. Model, sinüzoidal fonksiyonları kullanarak, eğitim setindekilerden daha uzun olan daha önce görülmemiş dizi uzunluklarına da ölçeklenebilir (Vaswani ve diğerleri, 2017: 6001).

1.6. OTOMATİK KONUŞMA TANIMA

Otomatik konuşma tanıma, elli yılı aşkın bir süredir aktif bir araştırma alanı olmuştur. Daha iyi insan-insan ve insan-makine iletişimini geliştirmede her zaman önemli bir köprü olarak görülmüştür. Geleneksel otomatik konuşma tanıma yöntemleri ve bunların uygulamaları organizasyonel gelişim için kritik bir hale geldikçe, bunların uçtan uca uygulamaları gelişim göstermeye başlamış ve bu konuda daha yenilikçi yöntemler geliştirilmeye başlanmıştır (Yu ve Deng, 2016).

1.6.1. Geleneksel Otomatik Konuşma Tanıma

GKKT, bir ses kaydının içindeki kelimeleri ve cümleleri doğru bir şekilde tanıma yeteneğidir. Bu teknoloji, bir ses kaydeden cihaz ve bir yazılım programı

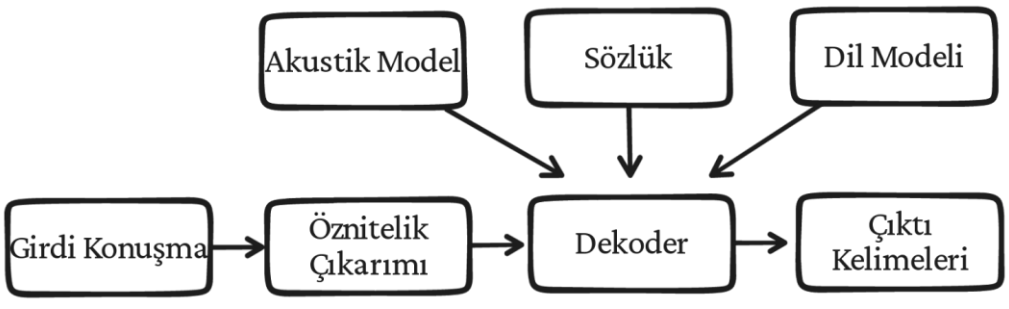
kullanarak çalışır. Yazılım programı, kaydedilen ses dosyasını analiz eder ve konuşmacının söylediklerini kelime kelime tanımaya çalışır.

GKKT, birçok farklı alanda kullanılabilir, ancak en yaygın olarak telefon hizmetlerinde veya bilgisayar yazılımlarında kullanılmaktadır. Örneğin, bazı telefon hizmetlerinde, kullanıcılar komutları sesleriyle verir ve GKKT yardımıyla, cihaz komutları anlar ve uygular (Rashmi, 2014: 5258). Ayrıca, GKKT teknolojisi, dil öğreniminde kullanılabilir ve konuşmacının telaffuz ve dil kurallarını düzgün bir şekilde uygulamasını sağlayabilir (Rashmi, 2014: 5258).

GKKT teknolojisi, giderek daha da gelişmekte ve günümüzde, yüksek doğruluk oranlarına sahip olmaktadır. Ancak, GKKT teknolojisi hala bazı sınırlamalara sahiptir ve bazı durumlarda yanlış kelime tanıma gibi hatalar yapabilir (Rashmi, 2014: 5259). Bu nedenle, GKKT teknolojisi kullanılırken, kullanıcıların dikkatli olmaları ve teknolojinin sınırlarının farkında olmaları önemlidir.

Geleneksel ASR sistemleri, Şekil 6'da gösterildiği gibi ayrı bileşenler içerir. Özellik çıkarıcı, sesi özellik vektörlerine veya gözlemlere dönüştürür. Gözlemler, fonem dizilerine bağlı gözlem olasılıklarını sağlayan bir akustik modele beslenir. Dil modelleme bileşeni, P(W) kelime dizileri üzerinden apriori olasılık dağılımını hesaplar. Sözlük veya telaffuz sözlüğü, fonem dizilerini kelimelerle eşleştirmek için kullanılır, bu da bir posteriori olasılık dağılımı P(O|W) ile sonuçlanır.

Şekil 6: Geleneksel ASR Boru Hattı.



Kaynak: Yazar tarafından derlenmiştir

1.6.2. Uçtan Uca Otomatik Konuşma Tanıma

UUTK, bir konuşmacının söylediği kelimeleri ve cümleleri doğru bir şekilde tanıma yeteneğidir. Bu teknoloji, bir konuşmacının sesini kaydeden bir cihaz ve bir yazılım programı kullanarak çalışır. Yazılım programı, kaydedilen ses dosyasını analiz eder ve konuşmacının söylediklerini doğru bir şekilde kelime kelime tanımaya çalışır.

UUTK teknolojisi, geleneksel otomatik konuşma tanıma teknolojisinden farklı olarak, konuşmacının söylediği kelimeleri ve cümleleri doğrudan doğruya tanımaya çalışır. Yani, UUTK teknolojisi, konuşmacının söylediği kelimeleri ve cümleleri ayrı ayrı tanımaya çalışmaz; aksine, konuşmacının söylediği kelimeleri ve cümleleri bütünüyle bir bütün olarak tanımaya çalışır. Bu, UUTK teknolojisinin doğruluk oranını artıran ve kullanımını kolaylaştıran önemli bir özelliktir.

UUTK teknolojisi, birçok farklı alanda kullanılabilir. Örneğin, UUTK teknolojisi, GTTK'de olduğu gibi, dil öğreniminde kullanılabilir ve konuşmacının telaffuz ve dil kurallarını düzgün bir şekilde uygulamasını sağlayabilir (Rashmi, 2014: 5260). Ayrıca, UUTK teknolojisi, işitme engelliler için önemli bir yardımcı teknolojidir ve bu kişilerin iletişim kurabilmelerini sağlar (Rashmi, 2014: 5260).

UUTK teknolojisi, giderek daha da gelişmekte ve günümüzde, yüksek doğruluk oranlarına sahip olmaktadır. Ancak, UUTK teknolojisi hala bazı sınırlamalara sahiptir ve bazı durumlarda yanlış kelime tanıma gibi hatalar yapabilir (Rashmi, 2014: 5261). Bu nedenle, UUTK teknolojisi kullanılırken, kullanıcıların dikkatli olmaları ve teknolojinin sınırlarının farkında olmaları önemlidir. Bu teknolojinin kullanımı, konuşmacının sesini doğru bir şekilde kaydetmesine ve yazılım programının güncel olmasına bağlıdır. Ayrıca, UUTK teknolojisi kullanılırken, konuşmacının sesini yüksek bir kalitede kaydetmesi ve konuşma hızının düzgün bir şekilde ayarlanması önemlidir (Rashmi, 2014: 5261).

UUTK teknolojisi, gelecekte daha da gelişecek ve kullanımı daha da yaygınlaşacaktır. Özellikle, dil öğrenimi, işitme engellilerin iletişimine yardımcı olma ve telefon hizmetleri gibi alanlarda UUTK teknolojisinin kullanımı daha da artacaktır. Bu teknolojinin gelişimi, insanların iletişimini kolaylaştıracak ve birçok farklı alanda önemli bir rol oynayacaktır.

1.6.3. Çağrı Merkezlerinde Otomatik Konuşma Tanıma Kullanılması

Çağrı merkezlerinde ASR sistemleri, bir konuşmacının söylediği kelimeleri ve cümleleri doğru bir şekilde tanıma yeteneğine sahip olan teknolojilerdir. Bu teknolojiler, bir konuşmacının sesini kaydeden bir cihaz ve bir yazılım kullanarak çalışır. Yazılım, kaydedilen ses dosyasını analiz eder ve konuşmacının söylediklerini doğru bir şekilde kelime kelime tanımaya çalışır.

Çağrı merkezlerinde ASR sistemleri, kullanıcıların birçok farklı senaryoda kullanılabilir. Örneğin, bir kullanıcı, çağrı merkezine bir telefon numarası aramak suretiyle bağlanabilir ve ASR sistemi yardımıyla, kendisiyle ilgili bilgileri söyleyerek bir işlem yapabilir (Zweig ve diğerleri, 2006: 1). Ayrıca, bir kullanıcı, bir web sitesinden çağrı merkezine bağlanabilir ve ASR sistemi yardımıyla, kendisiyle ilgili bilgileri girerek bir işlem yapabilir (Seghal ve Raj, 2018: 26).

Çağrı merkezlerinde ASR sistemlerinin kullanımı, organizasyonlar için birçok fayda sağlar. Örneğin, ASR sistemleri, çağrı merkezlerinde çalışan işgücünün verimliliğini artırabilir ve işlemlerin hızını arttırabilir (Zweig ve diğerleri, 2006: 1). Ayrıca, ASR sistemleri, çağrı merkezlerinin kullanımını kolaylaştırabilir ve kullanıcıların deneyimini iyileştirebilir (Zweig, 2006: 2; Seghal ve Raj, 2018: 27).

Bu, çağrı merkezlerinde daha fazla işlem yapılabilmesini ve daha fazla müşteriye hizmet verilebilmesini sağlar. Müşterilerin çağrı merkezlerine tekrar tekrar bağlanma ihtimalleri azalarak, organizasyonların müşteri memnuniyetini arttırmasını sağlar.

Çağrı merkezlerinde ASR sistemlerinin kullanımı bazı dezavantajları da beraberinde getirir. Örneğin, ASR sistemleri, bazı durumlarda yanlış kelime tanıma gibi hatalar yapabilir (Yi ve diğerleri, 2022: 2). Bu, kullanıcıların deneyimini olumsuz etkileyebilir ve müşteri memnuniyetini düşürebilir. Bu nedenle, çağrı merkezlerinde ASR sistemlerinin kullanımı, organizasyonlar tarafından dikkatli bir şekilde değerlendirilmelidir. Önemli olan, ASR sistemlerinin sağladığı faydaların ve dezavantajların dengeli bir şekilde değerlendirilmesidir.

Organizasyonlar, kendi iş süreçlerini ve müşteri ihtiyaçlarını dikkate alarak, en uygun ASR sistemini seçmelidir. Bu sistemlerin seçiminde, teknolojinin doğruluk oranı, kullanım kolaylığı ve maliyet gibi faktörler de dikkate alınmalıdır. Ayrıca,

organizasyonlar, ASR sistemlerinin kullanımının yanı sıra, insan gücünü de kullanarak, müşteri memnuniyetini artırmak için çaba sarf etmelidir.

1.6.4. Organizasyonel Gelişimde Otomatik Konuşma Tanıma Sistemlerinin Rolü

ASR sistemleri, organizasyonların gelişiminde önemli bir rol oynamaktadır. Özellikle son yıllarda, bu sistemlerin kullanımı artmış ve bu sistemler sayesinde organizasyonlar, işlerini daha verimli ve hızlı bir şekilde yapabilmektedir.

Birçok çalışmada, ASR sistemlerinin organizasyonlarda iş verimliliğini artırdığı ve iş süreçlerini hızlandırdığı gösterilmiştir. Örneğin, Filippidou ve Moussiades, Pallett gibi araştırmacılar, ASR sistemlerinin kullanımı ile birlikte iş verimliliğinin artışı ve iş süreçlerinin hızlandığını irdelemişlerdir (Filippidou ve diğerleri, 2020: 73; Pallett, 2003: 484).

Ayrıca, ASR sistemleri sayesinde organizasyonlar, müşteri hizmetlerini de geliştirebilmektedir. Bu sistemler sayesinde müşteriler, kolayca ve hızlı bir şekilde sorunlarını çözebilmektedir. Örneğin, Song ve diğerleri çalışmalarında ASR sistemlerinin kullanımı ile birlikte müşteri memnuniyetinin arttığını belirtmişlerdir (Song ve diğerleri, 2022: 6952).

Sonuç olarak, ASR sistemleri organizasyonların gelişiminde önemli bir rol oynamaktadır. Bu sistemler sayesinde iş verimliliği artmakta ve iş süreçleri hızlanmaktadır. Ayrıca, müşteri hizmetleri de gelişebilmektedir. Bu nedenle, ASR sistemlerinin kullanımının önemi giderek artmaktadır.

İKİNCİ BÖLÜM
VERİ, YÖNTEM VE KISITLAR

2.1. VERİ SETİ

Çalışmada veri seti olarak Mozilla Common Voice kullanılmıştır. Common Voice veri seti, konuşma teknolojisi araştırma ve geliştirilmesine yönelik, çok dilli, yazıya dökülmüş bir konuşma koleksiyonudur. Common Voice, ASR algoritmaları geliştirilmesi için tasarlanmış ancak, dil tanımlama gibi diğer alanlarda da yararlı olabilmektedir. Ölçek ve sürdürülebilirlik elde etmek için Common Voice projesi hem veri toplama hem de veri doğrulama için gönüllü kullanıcılardan oluşan, belirli metinleri okuyup ses dosyayı olarak kaydeden bir kitleyi kaynak olarak kullanmıştır.

Bu çalışmada kullanılan sürüm 6.1, 76 dili içermekte ve Kasım 2021 itibariyle veri toplanan toplam 85 dili barındırmaktadır. Şu ana kadar 80.000'den fazla kişi gönüllü olarak bu veri setinin oluşturulması çalışmasına katılmış ve bunun sonucunda 13.905 saatlik kaydedilmiş, 11.192 saatlik ise doğrulanmış veri elde edilmiştir. Türkçe dili için ise 44 saatlik kaydedilmiş ve 39 saatlik ise doğrulanmış veri bulunmaktadır. Tercih edilen veri seti, açık kaynak veri setleri içerisinde hem saat sayısı hem de dil sayısı açısından konuşma tanıma için alanındaki en büyük ses-metin eşleşmelerinden birini oluşturmaktadır.

Tablo 1'de görüldüğü üzere veri setinde zaman içerisinde toplanan verilere ait öznitelikler bulunmaktadır. Burada, veri setinde toplanan erkek sesleri ile 19-39 yaşları arasındaki kişilerin sesinde bir yoğunluk görülmektedir. Bu da çalışmada kullanılan veri setinin bu yaş aralığı ve erkek sesleri içerisinde en yüksek doğruluğa ulaşabileceğin önyargısını doğurmaktadır. Fakat erişim kolaylığı, açık kaynaklı oluşu ve saat sayısı göz önünde bulundurulduğunda en kullanışlı veri seti olduğu için çalışmada tercih edilmiştir.

Tablo 1: Mozilla Comon Voice Corpus Türkçe Veri Setinin Özniteliklerinin Zaman İçerisinde Değişimi.

Tarih	**Kayıtlar**			**Cinsiyet**		
21.07.21	Onaylı Saatler	Toplam Saatler	Özgün Ses	Erkek	Kadın	N/A
	30	37	960	68%	6%	26%
			Yaş Bandı			
19-29	30-39	40-49	50-59	60-69	>70	N/A
47%	17%	2%	4%	1%	0%	26%

Kaynak: (Mozilla Common Web Sitesi)

Common Voice projesi 2017 yılının Temmuz ayında İngilizce diline odaklanılarak başlatılmış, daha sonra ise Hazian 2018 yılında, diğer diller için de kullanıma sunulmuştur. Common Voice, yapısı itibarı ile sürdürülebilir bir veri toplama hattına sahiptir ve toplanan veriler çapraz doğrulama ile kontrol edilmektedir. Bu kontrol sistemi basitçe bir kullanıcı tarafından okunan metnin diğer kullanıcılar tarafından doğru şekilde okunup okunmadığının oylanması ile gerçekleştirilir. Her bir ses klibini en fazla üç katılımcı dinlemektedir. Doğru oy sayısı yanlış oy sayısından en az 2 adet fazla olmayan metin okumaları doğrulanmamış veri olarak arşivlenirken, diğer veriler doğrulanmış veri olarak arşivlenmektedir. Her bir dil için doğrulanmış eğitim, doğrulama ve test veri setleri en az 2 adet olumlu doğrulama ile oylanmış veriler kullanılarak elde edilir. Bu veri setleri oluşturulurken ayrıca tekrar eden metinler bu veri setlerinden kaldırılarak çoklama önlenmiş olur. Oluşturulan ses klipleri 48kHz hızına sahip tek kanallı, 16bit MPEG-3 dosyaları olarak kaydedilir. WAV, FLAC gibi formatlara göre daha yüksek ses kaybı içeren bu formatın tercih edilmesinin nedeni Common Voice veri toplama platformunun web tabanlı arayüzünün performans tercihlerinden kaynaklanmaktadır. Ayrıca verilerin bu formatta saklanmasının bir diğer nedeni de diğer platformlar içerisinde de sıkça kullanılan bir format olan MPEG-3'ün yaygınlığından kaynaklanmaktadır.

Kullanıma sunulan dillerin her biri, Mozilla Common Voice'a ait web sitesinden sıkıştırılmış bir dizin olarak tek tek indirilebilir. Dizin, Sekmeyle Ayrılmış

Değerlere sahip altı dosya (yani TSV dosyaları) ve bunların tümünü içeren tek bir klipler alt dizini içerir. Altı TSV dosyasının her biri, ses verilerinin farklı bir bölümünü temsil eder ve altısı da şu sütun başlıklarına sahiptir: [client_id, path, cümle, olumlu oylar, olumsuz oylar, yaş, cinsiyet, aksan]. İlk üç sütun, konuşmacı için anonimleştirilmiş bir kimliğe, ses dosyasının konumuna ve okunan metne atıfta bulunur. Sonraki iki sütun, dinleyicilerin <audio,transscript> çiftini nasıl değerlendirdiği hakkında bilgi içerir. Son üç sütun, isteğe bağlı olarak sesin konuşmacısı tarafından bildirilen demografik verileri temsil eder.

2.2. YÖNTEM

2.2.1. Kendinden Denetimli Akustik Modelleme

SSAM, insan açıklamalı veriler gerektirmeyen bir öğrenme yöntemidir. Bazen denetimsiz öğrenmenin bir biçimi olarak kabul edilir. Bununla birlikte, denetimsiz öğrenme tipik olarak, örneğin kümeleme gibi belirli veri kalıplarına odaklanır. Buna karşılık, kendi kendini denetleyen öğrenme algoritmaları, genellikle verilerin herhangi bir gruplandırılmasını veya kümelenmesini gerçekleştirmez. Bunun yerine, kendi kendine denetimli öğrenmede, etiketlenmemiş verilerden bazı denetimli öğrenme görevleri otomatik olarak oluşturulur. Genel olarak, aynı girdinin görünmeyen bir parçası verilen girdi verilerinin gizli veya eksik kısımlarını veya özelliklerini kurtarmayı amaçlar. Bu tür görevleri çözmek için makine, verinin anlamını veya yapısını taşıyan güçlü temsilleri öğrenmeye zorlanır. Bu öğrenilmiş temsillerin, genellikle birkaç etiketle ince ayar yapıldıktan sonra, çeşitli aşağı akış görevlerinde faydalı olması beklenir.

Bu konuda önemli bir referans olarak Baevski ve diğerlerinin (2020) çalışması kabul edilebilir. Bu makalede, yazarlar kendinden denetimli eğitim yöntemini kullanarak bir ses tanıma sistemi geliştirmeyi amaçlamışlardır. Makalede, yazarlar geliştirdikleri yöntemi farklı ses veri kümeleri üzerinde test etmiş ve bu yöntemin performansını diğer benzer yöntemlerle karşılaştırmışlardır. Sonuç olarak, yazarlar kendinden denetimli eğitim yönteminin diğer benzer yöntemlerden daha iyi sonuçlar verdiğini göstermişlerdir (Baevski ve diğerleri, 2020; 1).

Ayrıca, Chi ve diğerlerinin (2021) bir diğer çalışmasında da kendinden denetimli akustik modelleme konusunda ayrıntılı bilgi sunmaktadır. Bu makalede, yazarlar kendinden denetimli eğitim yöntemini ses tanıma sistemlerinde kullanarak, ses verilerinde bulunan doğal özellikleri keşfederek ağın eğitimini gerçekleştirmişlerdir. Makalede, yazarlar geliştirdikleri yöntemin performansını farklı ses veri kümeleri üzerinde test etmiş ve bu yöntemin diğer benzer yöntemlerden daha iyi sonuçlar verdiğini göstermişlerdir (Chi ve diğerleri, 2021; 334).

Son zamanlarda, çeşitli kendi kendini denetleyen konuşma temsil modelleri önerilmiştir, örneğin vektör nicemli varyasyonel otomatik kodlayıcı (VQ-VAE) (Van Den Oord ve Vinyals, 2017: 30; Baevski ve Schneider, 2019), Mockingjay (Liu ve diğerleri, 2020: 1) ve Audio ALBERT (Chi, 2021: 335). Bu çalışmada, derin akustik temsiller elde etmek için Wav2vec 2.0 mimarisi (Baevski ve diğerleri, 2020: 12449) kullanılmıştır.

Sonuç olarak, kendinden denetimli akustik modelleme bir ses tanıma sistemi geliştirmede kullanılan önemli bir yöntemdir. Bu yöntem, ses verilerinde bulunan doğal özellikleri keşfeder ve bu özellikleri kullanarak ağın eğitimini gerçekleştirir.

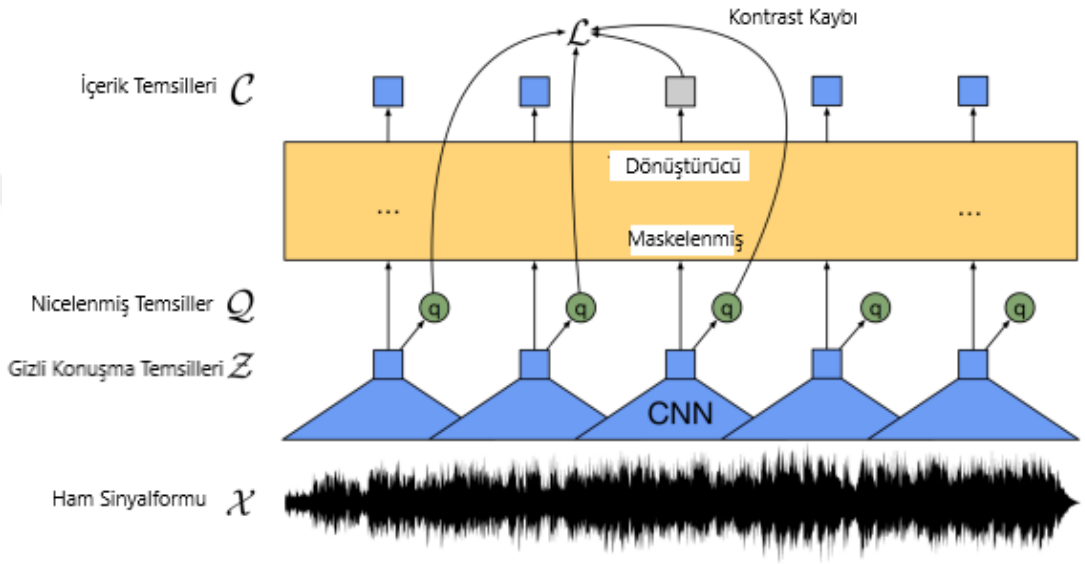
2.2.1.1. Wav2Vec 2.0

Wav2vec 2.0 birden çok evrişim katmanından ve dikkat katmanından oluşur. Bu yapı, son uçtan uca ASR modellerinde yaygın olarak kullanılmaktadır (Baevski, 2020: 12449). Evrişim katmanları algoritmaya girdi olarak verilen konuşmayı örneklem olarak alır ve daha sıkıştırılmış gizli temsilini üretir. Dikkat katmanının da kullanımı ile girdilerin nitelikli analizini yapabilir. Güçlü bağlam bağımlılığı modelleme yetenekleri, girdilerin, karşılaştırmalı eğitim sırasında modele doğru seçimleri yapması için güç verir.

Wav2Vec2'nin eğitim prosedürü iki aşamadan oluşur. İlk olarak, model, önemli miktarda etiketlenmemiş konuşma verisini kullanarak kendi kendini denetleyen ön eğitim sırasında akustik temsilleri öğrenir. İkinci aşamada, denetimli ince ayarda, model, ham sesten grafiklerin veya karakterlerin dizilerini doğru bir şekilde tahmin etmek için etiketlenmiş konuşma verileri üzerinde eğitilir.

Wav2Vec2'nin model yapısı Şekil 7'de gösterilmektedir. Ham konuşma dalga formu $x_i \in X$, ortalama ve birim varyansa normalleştirilir, f: $X \rightarrow Z$ özellik kodlayıcısına geçirilir ve gizli temsillere $z_i \in Z$ dönüştürülür. Wav2Vec2'deki özellik kodlayıcı, orijinal Wav2Vec ve VQ-Wav2Vec'deki tasarımını takip eder, ancak ancak içerisindeki aktivasyon fonksiyonları değiştirilir (Baevski, 2020: 12451).

Şekil 7: Wav2Vec 2.0 Çerçevesinin Çizimi.



Kaynak: (Baevski, 2020: 14450)

Wav2Vec2 ağı dönüştürücü tabanlıdır. Bir başka deyişle, konumsal kodlamadaki değişiklikler dışında BERT mimarisini takip eder. Burada, göreli konum bilgilerini öğrenmek için sabit konumsal gömmeler, göreli konumsal gömmelerle değiştirilir. Bu amaçla, Mohamed ve diğerlerinin (2019) çalışmalarına benzer şekilde dönüştürücü ağına bir evrişim katmanı eklenir (Mohamed ve diğerleri, 2019: 11660). Bağlam ağının farklı kurulumuna sahip iki model konfigürasyonu bulunmaktadır: Temel ve büyük baz modelleri 12 dönüştürücü bloğundan oluşmakta ve 768 model boyutuna, 3072 iç boyutuna ve 8 dikkat başlığına sahipken, geniş modellerinde model boyutu 1024 ve iç boyutu sırasıyla 1024 ve 4096'ya yükseltilmiş 16 dikkat başlığı ve 24 dönüştürücü bloğu bulunmaktadır.

Wav2Vec2'nin ikinci eğitim aşaması, aşağı akış görevi (ASR) için ince ayar, bir sınıflandırıcının rastgele başlatılmasıyla veya dönüştürücü ağının üstünde C

sınıflarına doğrusal bir katman projeksiyonu ile başlar. Sınıflar, karakterlerden ve bir kelime sınırı belirtecinden oluşan kelime dağarcığını temsil eder. Sınıflandırıcı, etiketli konuşma verileri üzerinde eğitilir ve standart bir CTC kaybıyla optimize edilir (Graves, 2006: 370). Özellik kodlayıcının ağırlıkları ince ayar sırasında dondurulur ve bu nedenle güncellenmez. Ayrıca, nicemleme modülü devre dışı bırakılır.

2.2.1.2. Kelime Hata Oranı

WER, bir konuşma tanıma veya makine çevirisi sisteminin performansının ortak bir ölçüsüdür (Heming ve diğerleri, 2000: 815). WER, sözcük seviyesinde tahminlenen sonuç ile mevcut sonuç arasındaki farklılığı hesaplamak için kullanılan bir metriktir. Kelime hata oranı daha sonra şu şekilde hesaplanabilir:

WER = (D + S + G) / N

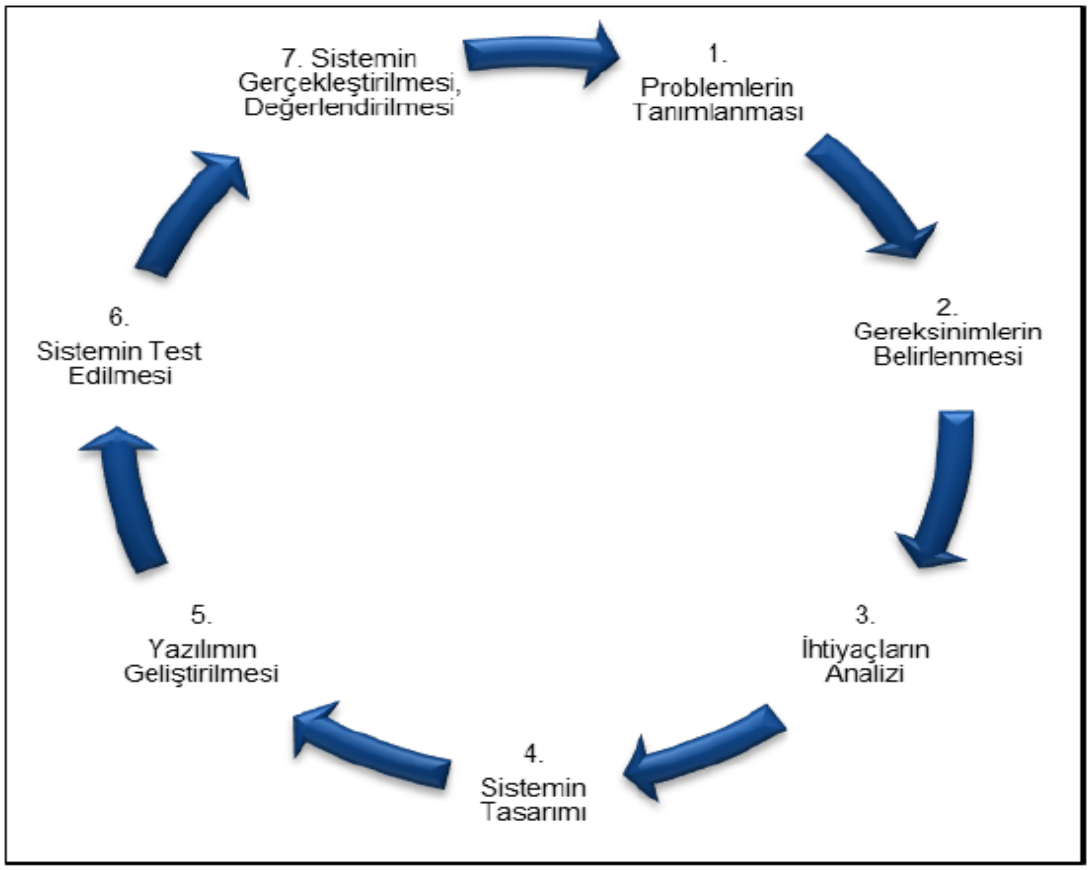
Burada; D, değişime maruz kalmış kelime sayısını, S, içerisinden karakter silinmiş kelime sayısını, G, içerisine karakter eklenmiş kelime sayısını, N ise ölçüme tabi tutulan toplam kelime sayısını ifade eder.

2.2.2. Sistem Geliştirme Yaşam Döngüsü

Sistem geliştirme yaşam döngüsü (SGYD), bir sistem ya da ürünün hayatı boyunca geçirdiği evrelerin tümünü ifade eder. Genellikle bu evreler; Kavramsal desen, tasarım, geliştirme, implementasyon, bakım ve destek, değerlendirme ve değişiklikler olarak tanımlanabilir. Sistem ya da ürünün performansı ve kullanıcı memnuniyeti değerlendirilerek, gerekli görülen değişiklikler yapılır. Bu evreler, bir sistem ya da ürünün hayatı boyunca tekrarlanabilir ve sürekli olarak iyileştirme ve geliştirme amacıyla yapılır (Laudon ve Laudon, 2014).

Bu yaklaşımlar, sistemlerin geliştirilmesinde analiz, tasarım, gerçekleştirme ve bakım aşamalarının planlanmasını ve koordinasyonunu sağlar. (Şekil 8) (Tecim, 2017; Laudon ve Laudon, 2014).

Şekil 8: Sistem Geliştirme Yaşam Döngüsü



Kaynak: (Tecim, 2017)

2.2.2.1. Problemlerin Tanımlanması

İhtiyaç duyulan yazılımın problemin kaynağına göre detaylandırılması ve analiz edilmesi problem için sunulacak olan çözümün yazılıma uygulanması sistemin tutarlılığını artıracaktır. Türkçe dili için konuşmadan metne dönüştürme süreçlerinde meydana gelen aksaklıklar için mevcut sistem ile getirilecek yenilik önerilerinde bulunulmuştur. Bu getirilecek yeniliğin bir çağrı merkezinde kullanılması planlanmıştır. Çağrı merkezinde kullanılması ile, gün içerisinde yapılan tüm görüşmelerin transkripsyonlarının otomatik olarak çıkartılması hedeflenmiştir. Mevcutta böyle bir sistem kullanılmayan bir çağrı merkezi organizasyonunda personelin performans ölçümlemesi ve müşteri memnuniyetinin dinamik ölçümlemesini yapabilmenin zorluğu problemimizin temelini oluşturmaktadır.

2.2.2.2. Bilgi Gereksinimlerin Belirlenmesi

Problem tanımlamalarının ardından konuşmanın metne dönüşümüne kadar geçen süreci bilişim tabanlı, nesnel, hızlı, web tabanlı platformda yazılımların oluşturulmasına karar verilmiştir. Yazılım platformlarının seçilmesine karar verilmesinde, mobilite ve kullanım kolaylığına bağlı olarak açık kaynaklı paylaşılan bir platformun seçilmesine neden olmuştur. Seçilen bu açık kaynaklı platform üzerinde oluşturulacak olan ASR sistemi çağrı merkezi operatörleri ve konuştukları kişileri, zaman etiketi de içerecek şekilde, sürekli olarak transkripsyon yapacaktır. Böylelikle organizasyonel gelişim için gereken metne dönüşmüş konuşma verileri kullanıcılar ve müşteriler özelinde oluşup arşivlenebilecektir.

2.2.2.3. Sistem İhtiyaç Analizi

İhtiyaç analizi tekniklerini kullanılarak belirli bir soruna çözüm olacak ihtiyacı somutlaştırmak gerekmektedir.

İhtiyaç belirleme teknikleri aşağıdaki Delphi Tekniği, Progel (Dacum) Tekniği, Gözlem, Görüşme, Ölçme Araçları olarak sınıflandırılabilir (Çetin, 2011). Bu proje kapsamında gözlem ve ölçme araçları teknikleri kullanılmıştır.

İhtiyaç belirleme tekniklerinden gözlem ve ölçme, bir sistem ya da ürünün geliştirilmesinde kullanılan yöntemlerdir. Bu teknikler, sistem ya da ürünün ihtiyaçlarının belirlenmesinde ve bu ihtiyaçları karşılamaya yönelik olarak planların yapılmasında kullanılır (Çetin, 2011).

Gözlem tekniği, bir sistem ya da ürünün kullanımı sırasında oluşan durumları inceleyerek ihtiyaçların belirlenmesinde kullanılır. Bu teknik sayesinde, sistem ya da ürünün kullanıcılarının davranışları ve bu davranışlardan kaynaklanan ihtiyaçlar incelenebilir. Örneğin, çağrı merkezinde aranan kitlede hâkim bir şive veya ağız mevcut ise geliştirilen otomatik konuşma tanıma modeli bu verilerle tekrar eğitilerek performansının geliştirilmesi sağlanabilir.

Ölçme tekniği ise, bir sistem ya da ürünün performansını ölçerek ihtiyaçların belirlenmesinde kullanılır. Bu teknik sayesinde, sistem ya da ürünün kullanımı sırasında oluşan veriler toplanarak, bu verilerin analiz edilerek sistemin ya da ürünün

performansının değerlendirilebilir. Örneğin, bir sosyal medya platformunun kullanımı sırasında oluşan veriler incelenebilir ve bu veriler kullanılarak platformun performansının optimize edilebilmesi için gerekli önlemler alınabilir.

Gözlem ve ölçme teknikleri, bir sistem ya da ürünün geliştirilmesinde önemli rol oynar. Bu teknikler sayesinde, sistem ya da ürünün ihtiyaçları belirlenebilir ve bu ihtiyaçları karşılamaya yönelik olarak planlar yapılabilir. Ayrıca, bu teknikler sayesinde sistem ya da ürünün performansının değerlendirilebilmesi ve iyileştirilmesi için gerekli önlemler alınabilir.

Bu çalışma için yapılan organizasyonel gelişimde otomatik konuşma tanıma sistemi kullanılması aşamasında, çağrı merkezi örneği üzerinden yola çıkarak incelendiğinde, müşteriler ve operatörler için iki ayrı ASR sisteminin kullanılması gerektiği düşünülerek modelleme yapılmıştır. Elde edilen konuşma metinleri, anahtar kelime analizlerine tabi tutulmak üzere kuruluşa ait veri tabanında saklanacaktır. Böylece yöneticiler, müşteri eğilim ve analizlerini yaparken aynı zamanda çağrı merkezi personelinin müşteri ile iletişim kurma becerisi, teknik ve sosyal yeterliliklerini de ölçümleyebileceklerdir.

2.2.2.4. Sistem Tasarımı

Yazılım gereksinim analizinin yapılmasındaki amaç; kullanıcı, ürün ve ürün bileşeni gereksinimlerinin analizi ve üretilmesidir. Gereksinim türleri; kullanıcı ve fonksiyonel gereksinimler, mimari ve yapısal gereksinimler başlıkları adı altında tartışılmıştır.

Kullanıcı gereksinimleri: Kullanıcı gereksinimleri mevcut sistemde yapılamayan ve yapıldığında kullanıcıların iş yapış hızlarını arttıracak şekilde belirlenmiş olup sistemin kullanım alanlarını genişletecek ve sistemi etkin bir şekilde tamamlayacak şekilde belirlenmiştir.

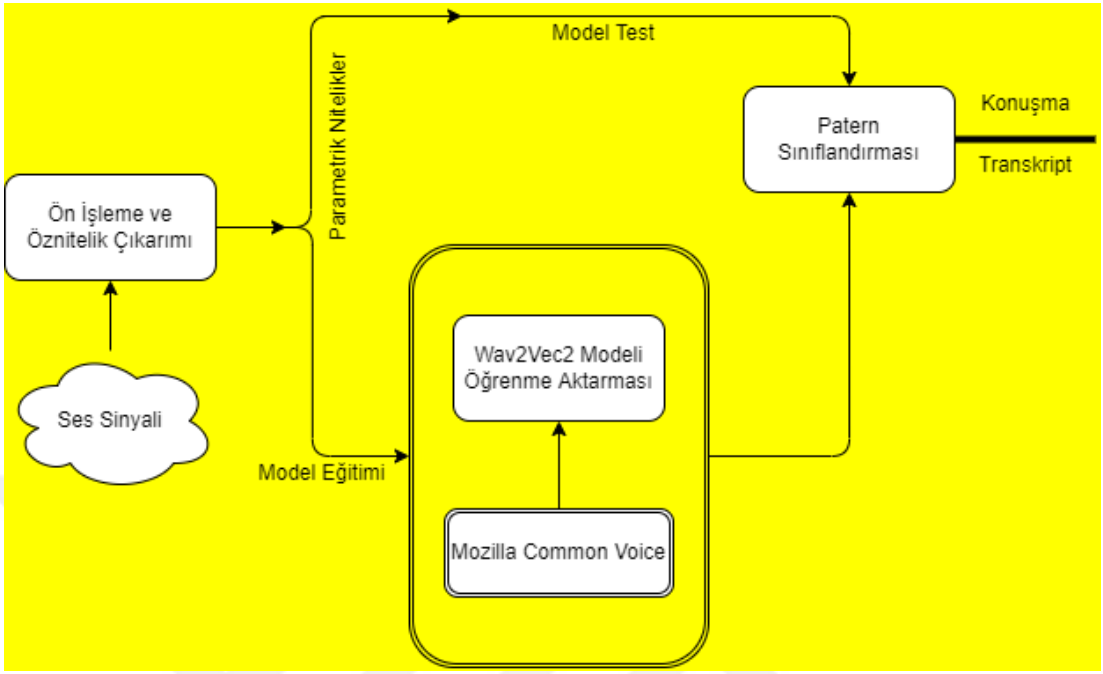
Fonksiyonel Gereksinimler: Fonksiyonel gereksinimler sistemin yeteneklerini detaylandırmak ve sistemde neyin nasıl yapılacağı konusunu yapısal ve işlevsel olarak incelemektedir. Ayrıca sistemin yetenekleri ve sistem altyapısının belirlenmesini sağlamaktadır. Hata yönetimi ile kullanıcı yazılım etkileşimi ele alınır.

Mimari ve Yapısal Gereksinimler: Şekil 9'da gösterilen sistem yazılım mimarisinde kullanılan birimlerin arasında döngüsel bir iletişim söz konusudur. Veri kaynağından çekilen ve Wav2Vec2.0 eğitiminde kullanılan sinyal ve metin verileri ile oluşturulan model yine test için elde edilen veriler ile test edildikten sonra başarımı yüksek olan model ile otomatik konuşma tanıma işlemleri gerçekleştirilmektedir. Yapısal olarak ise Python programlama dili ve bu dile ait olan açık kaynaklı tensorflow, wandb, huggingface-hub, soundfile, jiwer, typeguard, tensorflow_hub kütüphanelerine gereksinim vardır.

Şekil 9'da proje kapsamında geliştirilen yazılım mimarisi gösterilmektedir. Şekil 10'da ise geliştirilen modelin bir çağrı merkezi içerisinde örnek bir kullanım senaryosu görülmektedir. Bu senaryo her bir çağrı özelinde tekar gerçekleşen transkripsyonların oluştuğu ve ilgili veri depolama alanında saklandığı kaynak yapıyı göstermektedir. Sürecin en yalın ve organizasyonel ihtiyaca göre adapte edilebilir olması için her kullanım senaryosu için aynı şekilde uygulanabilecek olan mimari çizimler yapılmıştır. Bir üst düzeydeki mimari tasarımlar organizasyonların kullandığı çağrı merkezi yazılımları, operatör sistemleri ile değişiklik gösterebilir.

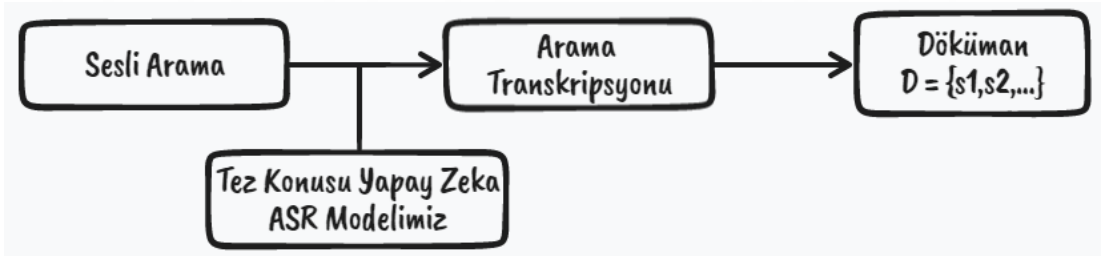
Otomatik Konuşma Tanıma sisteminin entegrasyonu sonrasında çağrı merkezinde kullanılması senaryosunda ise, çağrı merkezi süpervizörünün çağrı merkezi çalışanlarının yarısının konuşmalarına anlık erişim sağlaması ve onlara çağrı esnasında veya sonrasında müdehale etmesi ile birlikte çağrı merkezi çalışanlarının aylık bazda gerçekleştirilen başarılı görüşme sayısının toplam görüşme sayısına oranı hesaplanacaktır. Hesaplanan bu orandaki değişiklik otomatik konuşma tanıma sisteminin entegrasyonunun organizasyonel gelişime etkisi ile ilgili olarak somut etkisinin hesaplanmasını mümkün kılacaktır.

Şekil 9: Sistem Yazılım Mimarisi – 1



Kaynak: Yazar tarafından derlenmiştir.

Şekil 10: Sistem Yazılım Mimarisi- 2



Kaynak: Yazar tarafından derlenmiştir.

Beklentileri karşılayan, zaman planına uygun bir şekilde ilerleyen ve bütçe olarak planlanmış bütçe ile korele bir şekilde ilerleyen, tüm bu şartları sağlarken tolere edilebilir bir hata ile bunları gerçekeleştirebilen projeler başarılı proje olarak tanımlanmaktadır (Kan, 2003).

Projelerin yaşam döngüleri sırasında ortaya çıkan hatalar, proje takvimi çerçevesinde tamamlanma süresini uzatabilir. Bu tür istenmeyen durumlarla baş etmek için, projenin tekrar tekrar incelenmesi önemlidir. Testler yardımıyla bu hataların

tespit edilmesi ve düzeltilmesi, ileriki aşamalarda ortaya çıkabilecek maliyetleri azaltarak projeyi daha verimli hale getirir (Kan, 2003).

2.2.2.5. Yazılımın Geliştirilmesi

Uygulama tasarlanırken kullanım kolaylığı ve açık kaynaklı erişim açısından geliştirilen yapay zekâ modeli huggingface.co/emre alt sekmesi altında, model dosyaları paylaşılmıştır. Burada huggingface hub üzerinden API ile iletişim kurarak geliştirilen modele aktif olarak sorgu gönderilmesinin mümkün olması sağlanmıştır. Yapay zekâ modelinin geliştirilmesi için Google Colab ortamı tercih edilmiştir. Burada sağlanan ücretsiz GPU modelin kısa sürede gerçeklenebilmesini sağlamıştır.

2.2.2.6. Test

Gerçek dünyada, eğitilmiş bir yapay zekâ algoritmasının sonuçları her zaman eğitim veri setindeki performans ile eşleşmez. Uygulamanın eğitimi tamamlandıktan sonra model tarafından hiç görülmemiş veriler ile modelin test edilmesi gerekir. Bu test işlemleri için modelin önceden belirlenmiş performans metrikleri olması gerekmektedir. Wav2Vec2 için bu çalışmada kullanılan performans metrikleri sırası ile kayıp değeri, kelime hata oranı, karakter hata oranı metrikleri kullanılmıştır. Kayıp fonksyonu Wav2Vec2 bölümünde açıklanmıştır. Kelime hata oranı ve karakter hata oranı ise birim gönderi içerisindeki hatalı çevirinin doğru çeviriye oranını ifade etmektedir.

2.3. KISITLAR

Eğitim veri seti Mozilla Common Voice veriseti kullanılarak gerçekleşmiştir. Veri setinde kadın-erkek sesleri ve yaş dağılımları normal dağılım göstermemektedir. Bundan dolayı geliştirilmiş olan model en yüksek başarımı dijital katkısı en yüksek olan genç ve orta yaşlı erkek kitle içerisinde göstermektedir. Çocuklar, yaşlılar ve kadınlarda ise göreceli olarak daha düşük bir performans elde edilmektedir. Bu nedenle modelin en yüksek performansında göreceli de olsa bir kısıtlama mevcuttur.

ÜÇÜNCÜ BÖLÜM
UYGULAMANIN GERÇEKLENMESİ

3.1. UYGULAMA YAZILIMININ GERÇEKLENMESİ

Uygulamanın tamamı için Python programlama dili tercih edilmiştir. Açık kaynaklı ve ücretsiz olarak kullanılabilen python geliştirici Google colab platformu üzerinden programlama gerçekleştirilmiştir. Programlama ile ilgili bütün aşamaları detaylı ve açıklayıcı bir şekilde anlatan bir Google Colab not defteri yazar tarafından deneyin tekrarlanabilirliği için hazırlanmış ve bu çalışmanın eki olarak sunulmuştur. Hazırlanan bu not defterinde uçtan uca modelin eğitim aşamaları detaylı bir şekilde anlatılmıştır.

3.2. VERİ, JETONLAR, ÖZNİTELİK SETLERİNİN OLUŞTURULMASI

ASR modelleri konuşmayı metne dönüştürür; bu, konuşma sinyalini modelin giriş formatına işleyen bir özellik çıkarıcıya, mesela bir özellik vektörüne ve modelin çıktı formatını metne dönüştüren bir belirtecine ihtiyacımız olduğu anlamına gelir.

Dönüştürücü yapısın'da, Wav2Vec2 modeline verilerin tokenlara ayrılması için Wav2Vec2 CTC Tokenizer adlı bir belirteç modülü ve öznitelik çıkarımı için ise Wav2Vec2 Feature Extractor adlı bir özellik çıkarıcı eşlik eder. Türkçe için önceden hazırlanmış olan Mozilla Common Voice veriseti yüklenerek çalışma başlatılır. Şekil 11'de kullanılan kod ile bu işlem gerçekleştirilir.

Şekil 11: Veri Setinin İndirilmesi



Kaynak: Yazar tarafından derlenmiştir.

Mozilla Common Voice veri seti sadece konuşma verisi ve metin karşılığını barındırmaz. Bu çalışma için aksan, yaş, cinsiyet, segment ve doğrulama sayısı verilerini model girdi veri setinden çıkartarak konuşma verisi ve metin karşılığını bırakacak şekilde veri setini azaltmaya gidilir. Şekil 12'te bununla ilgili veri manipülasyonu görülmektedir.

Şekil 12: Kullanılmayan Verilerin Ayıklanması ve Veri Setinden Örnekler

```
[ ] common_voice_train = common_voice_train.remove_columns(["accent", "age", "client_id", "down_votes", "gender", "locale", "segment", "up_votes"])
    common_voice_test = common_voice_test.remove_columns(["accent", "age", "client_id", "down_votes", "gender", "locale", "segment", "up_votes"])
```

```
show_random_elements(common_voice_train.remove_columns(["path", "audio"]), num_examples=10)
```

	sentence
0	Bir vergi politikası, nasıl bir yapıda ve ne düzeyde vergi gelirleri hedeflemelidir?
1	Bal!
2	İlk BM barış gücü grubu bölgeye geldi.
3	Onay, iki gün süren tartışmalar sonrasında geldi.
4	Bu da fiyatarda düşüşe yol açacak.
5	"Lajiç'in eşi de iş ihtiyacını vurguluyor."
6	Çantaları taşıyamayacakları kadar ağırdı.
7	Kısıtlı kaynaklar tedavi kalitesini de etkiliyor.
8	Ancak Yunanlılar işini biliyordu.
9	Kosova mutfağı kara iklimiyle son derece uyumlu.

Kaynak: Yazar tarafından derlenmiştir.

Metin karşılıklarında, `,.?!;:` gibi bazı özel karakterler içerdiğini görebiliriz. Bir dil modeli olmadan, konuşma parçalarını bu tür özel karakterlere göre sınıflandırmak çok daha zordur çünkü bunlar gerçekten karakteristik bir ses birimine karşılık gelmezler. *Örneğin*, `"s"` harfinin az çok net bir sesi vardır, oysa özel karakter `"."` yoktur. Ayrıca bir konuşma sinyalinin anlamını anlamak için genellikle transkripsiyona özel karakterler eklemek gerekli değildir. Ek olarak, metni yalnızca küçük harflere sahip olacak şekilde normalleştiririz ve sonuna bir sözcük ayırıcı belirteci ekleriz. Şekil 13'te, özel karakterlerin çoğundan ayrıştırılmış ve küçük harfler ile normalleştirilmiş metin örnekleri görülmektedir.

Şekil 13: Ön İşleme Tabi Tutulmuş Transkripsyon Verileri

	sentence
0	drugclan aynı fikirde değil
1	iktidara geldiğinde tamamen farklı olacak
2	tutuklanan doktorların sayısı her hafta artıyor
3	anıtın yeri henüz belirlenmiş değil
4	proje iki bin on yılında tamamlanacak
5	ilk fırın altı aydır atıl durumdaydı
6	dimitrov isim önerisi getirilmediğini doğruladı
7	reform süreçlerinin hızından memnun musunuz
8	etrafımda bir sürü yerli insan var
9	bu deliği de tıkadılar

Kaynak: Yazar tarafından derlenmiştir.

Bu hali ile oluşan verisetimizdeki özgün karakterlerin listesi {' ': 13, '"': 10, 'a': 14, 'b': 15, 'c': 22, 'd': 26, 'e': 11, 'f': 12, 'g': 18, 'h': 17, 'i': 25, 'j': 33, 'k': 20, 'l': 5, 'm': 6, 'n': 28, 'o': 24, 'p': 4, 'q': 23, 'r': 35, 's': 32, 't': 7, 'u': 30, 'v': 0, 'w': 19, 'x': 9, 'y': 34, 'z': 27, 'â': 36, 'ç': 21, 'ë': 1, 'î': 31, 'ö': 16, 'ü': 37, 'ğ': 3, 'ı': 2, 'ş': 29, '': 8} şeklinde olmuştur. Alfabenin tüm harfleri veri kümemizde bulunmaktadır ve ayrıca “ “ ve ‘ özel karakterleri de çıkarılmıştır. Bu özel karakterleri hariç tutmadık çünkü; model, bir kelimenin ne zaman biteceğini tahmin etmeyi öğrenmelidir, yoksa model tahmini her zaman kelimeleri birbirinden ayırmayı imkansız kılacak bir karakter dizisi olacaktır. Ayrıca yukarıdaki transkripsiyonlardan veri setinde 'Musa'nın' gibi kesme işareti içeren kelimelerin olduğu anlaşılıyor, bu yüzden kesme işareti veri setinde tutulmuştur. Modelin "a" ve "A" arasında ayrım yapmasını istemiyoruz. "a" ve "A" arasındaki fark, hiçbir şekilde harfin "sesine" bağlı değildir, daha çok dilbilgisi kurallarına bağlıdır. Bu nedenle, büyük ve büyük olmayan harfler arasındaki farkı ortadan kaldırmak mantıklıdır, böylece modelin konuşmayı yazıya dökmeyi öğrenmesi daha kolay olur. `" "` Kendi belirteç sınıfına sahip olduğunu daha açık hale getirmek için, ona daha görünür bir `|` karakteri verilmiştir. Ayrıca, modelin daha sonra Common Voice'un eğitim setinde karşılaşılmayan karakterlerle başa çıkabilmesi için bir "bilinmeyen" belirteç de eklenmiştir. Son olarak, CTC'nin "boş belirtecine" karşılık gelen bir dolgu belirteci de ekleriz. Bu hali ile sözlüğümüz 40 özgün karaktere sahip olur.

Konuşma sinyallerinin bilgisayarlar tarafından ele alınabilmesi için önce örnekleme olarak adlandırılan ayrıklaştırılması gerekir. Örnekleme hızı burada önemli bir rol oynar, çünkü saniyede konuşma sinyalinin kaç veri noktasının ölçüldüğünü tanımlar. Bu nedenle, daha yüksek bir örnekleme oranıyla örnekleme, gerçek konuşma sinyalinin daha iyi bir şekilde yaklaştırılmasıyla sonuçlanır, ancak aynı zamanda saniyede daha fazla değer gerektirir.

Önceden eğitilmiş bir kontrol noktası, girdi verilerinin, üzerinde eğitildiği verilerle aşağı yukarı aynı dağıtımdan örneklenmesini bekler. İki farklı hızda örneklenen aynı konuşma sinyalleri çok farklı bir dağılıma sahiptir, Örnek olarak, örnekleme hızının iki katına çıkarılması, veri noktalarının iki kat daha uzun olmasına neden olur. Böylece, bir ASR modelinin önceden eğitilmiş bir kontrol noktasında ince ayar yapmadan önce, modeli önceden eğitmek için kullanılan verilerin örnekleme hızının, modelde ince ayar yapmak için kullanılan veri kümesinin örnekleme hızıyla eşleştiğini doğrulamak çok önemlidir. XLSR-Wav2Vec2'nin ses verileri üzerinde önceden eğitilmiştir. Bu veri kümelerinin çoğu 16 kHz'de örneklendiğinden ötürü 48 kHz'de örneklenen Common Voice, eğitim için 16 kHz'e indirilmelidir. Bu nedenle, ince ayar verilerimiz 16 kHz'e düşürülmüştür. Şekil 14'de bu dönüşüm için kullanılan kod görülmektedir.

Şekil 14: Öznitelik Dönüştürücü Fonksiyon

```
from transformers import Wav2Vec2FeatureExtractor

feature_extractor = Wav2Vec2FeatureExtractor(feature_size=1, sampling_rate=16000, padding_value=0.0, do_normalize=True, return_attention_mask=True)
```

Kaynak: Yazar tarafından derlenmiştir

Eğitim veri seti incelendiğinde, ses dosyasına ait vektörize edilmiş temsili, Şekil 15'te Common Voice veri seti içerisindeki orijinal ve dönüştürülmüş örneklem durumu görüntülenebilmektedir 48000Hz den 16000Hz'e düşürülen örneklem boyutu ve karşılık gelen metin çıktılarını bu proje ile birlikte paylaşılan not defterinden görüntülenebilir. Son olarak, verileri eğitim için 'Wav2Vec2ForCTC' tarafından beklenen formatta işlemek için 'Wav2Vec2Processor'dan yararlanılabilir. Bunun için `map(...)` fonksiyonunu kullanılmıştır. İlk olarak, sadece `batch["audio"]` çağırarak ses verilerini yükleyip yeniden örneklenmekte, ikinci olarak, yüklenen ses dosyasından "input_values" değerleri çıkartılmaktadır. Bu yalnızca normalleştirmeyi

içerir, ancak Log-Mel filtreleri gibi diğer konuşma modelleri için bu adım, çıkarma işlemine karşılık gelebilir. Üçüncü olarak, transkripsiyonları kimlikleri etiketlemek için kodlamak gerekmektedir. Bu eşleme işlevi, 'Wav2Vec2Processor' sınıfının nasıl kullanılması gerektiğine iyi bir örnek teşfkil etmektedir. "Normal" bağlamda, "processor(...)" çağrısı, "Wav2Vec2FeatureExtractor"ın çağrı yöntemine yönlendirilir. Ancak işlemciyi "as_target_processor" bağlamına çevirirken, aynı yöntem "Wav2Vec2CTCTokenizer"ın çağrı yöntemine yönlendirilir. Bununla ilgili yapılan işlemler de yine şekil 15'te görülmektedir.

Şekil 15: Orijinal ve Dönüşmüş Ses Dosyaları ve Transkripsyonları

```
common_voice_train[0]["audio"]
```

```
{'array': array([ 0.0000000e+00,  0.0000000e+00,  0.0000000e+00, ...,
       -8.8930130e-05, -3.8027763e-05, -2.9146671e-05], dtype=float32),
 'path': '/root/.cache/huggingface/datasets/downloads/extracted/05be0c29807a73c9b099873d2f5975dae6d05
 'sampling_rate': 48000}
```

```
common_voice_train = common_voice_train.cast_column("audio", Audio(sampling_rate=16_000))
common_voice_test = common_voice_test.cast_column("audio", Audio(sampling_rate=16_000))
```

```
rand_int = random.randint(0, len(common_voice_train)-1)

print("Target text:", common_voice_train[rand_int]["sentence"])
print("Input array shape:", common_voice_train[rand_int]["audio"]["array"].shape)
print("Sampling rate:", common_voice_train[rand_int]["audio"]["sampling_rate"])
```

```
Target text: rusya ülkeye dünya örgütünde destek vermişti
Input array shape: (79872,)
Sampling rate: 16000
```

```
def prepare_dataset(batch):
    audio = batch["audio"]

    # batched output is "un-batched"
    batch["input_values"] = processor(audio["array"], sampling_rate=audio["sampling_rate"]).input_values[0]

    with processor.as_target_processor():
        batch["labels"] = processor(batch["sentence"]).input_ids
    return batch
```

```
common_voice_train = common_voice_train.map(prepare_dataset, remove_columns=common_voice_train.column_names, num_proc=4)
common_voice_test = common_voice_test.map(prepare_dataset, remove_columns=common_voice_test.column_names, num_proc=4)
```

Kaynak: Yazar tarafından derlenmiştir

3.3. MODELİN EĞİTİMİ

Veriler, eğitim hattını kurmaya başlamaya hazır olması için işlenir. Çoğu NLP modelinin aksine, XLSR-Wav2Vec2, çıkış uzunluğundan çok daha büyük bir giriş uzunluğuna sahiptir. Örnek olarak giriş uzunluğu 50000 olan bir örneğin çıktı uzunluğu 100'den fazla değildir. Büyük girdi boyutları göz önüne alındığında, eğitim gruplarını dinamik olarak doldurmak modelin verimini arttırmak için tercih edilmiştir ve bu da tüm eğitim örneklerinin yalnızca en uzun örnekle doldurulması gerektiği anlamına gelir. Bu nedenle, XLSR-Wav2Vec2'in ince ayarını yapmak, aşağıda tanımlanacak olan özel bir dolgu veri harmanlayıcı gerektirir.

- Değerlendirme metriği: Eğitim sırasında model kelime hata oranı üzerinden değerlendirilmelidir. Buna göre bir "compute_metrics" işlevi tanımlanmalıdır.
- Önceden eğitilmiş bir kontrol noktası yükleme: Önceden eğitilmiş bir kontrol noktası yüklememiz ve eğitim için doğru şekilde yapılandırılması gerekmektedir. Buna kontrol noktasından başlayarak eğitimin ilerletilmesi adı verilmektedir. Bu işlem, modelin ön eğitimli modelden öğrenme aktarması gerçekleştirebilmesi için gerçekleştirilmektedir.

Öncelikle veri toplayıcı bir sınıf (class) tanımlaması yapılması gereklidir. Yaygın veri toplayıcıların aksine, bu veri toplayıcı "girdi_değerleri" ve "etiketler"i farklı şekilde ele almalı ve bu nedenle üzerlerindeki ayrı doldurma işlevlerine uygulanmalıdır (yine XLSR-Wav2Vec2'nin bağlam yöneticisini kullanır). Bu gereklidir çünkü konuşmada giriş ve çıkış farklı modalitelere sahiptir, yani aynı doldurma işlevi tarafından ele alınmamalıdırlar. Yaygın veri toplayıcılara benzer şekilde, `-100` ile etiketlerdeki dolgu belirteçleri, kayıp hesaplanırken bu belirteçler dikkate alınmaz.

Daha sonra XLSR-Wav2Vec2 kontrol noktası yüklenir ve eğitim ile ilgili tüm parametreler tanımlanır. Burda "group_by_length", benzer girdi uzunluğundaki eğitim örneklerini tek bir toplu iş halinde gruplayarak eğitimi daha verimli hale getirir. Bu, modelden geçirilen toplam işe yaramaz input tokenlarının sayısını büyük ölçüde azaltarak eğitim süresini önemli ölçüde hızlandırabilir. "Learning_rate" ve "weight_decay", ince ayar kararlı hale gelene kadar buluşsal olarak ayarlanmıştır. Bu

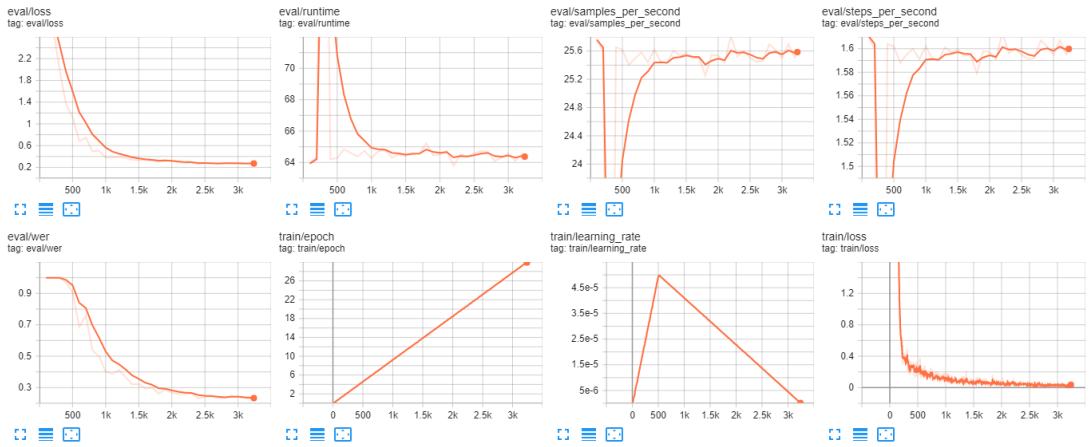
parametrelerin büyük ölçüde Common Voice veri kümesine bağlı olduğundan bu çalışma özelinde kullanılabilir haldedir.

Bu modelin eğitimi için kullanılan parametreler sırasıyla şu şekildedir: learning_rate: 0.0005, train_batch_size: 2, eval_batch_size: 8, seed: 42, distributed_type: multi-GPU, num_devices: 8, total_train_batch_size: 32, total_eval_batch_size: 16, optimizer: Adam with betas=(0.9,0.999) and epsilon=1e-08, lr_scheduler_type: linear lr,_scheduler_warmup_steps: 500, num_epochs: 30.0, mixed_precision_training: Native AMP. Bu parametreler ile 30 iterasyon eğitimden sonra modelin performans skorlarının sonuçları ve genel performans grafikleri ise Tablo 2 ve Şekil 16'da gösterilmektedir.

Tablo 2: Modelin Performans Sonuçları

Eğitim Kaybı	İterasyon	Adım	Doğrulama Kaybı	Kelime Hata Oranı (WER)
0.4388	3.7	400	1.366	0.9701
0.3766	7.4	800	0.4914	0.5374
0.2295	11.11	1200	0.3934	0.4125
0.1121	14.81	1600	0.3264	0.2904
0.1473	18.51	2000	0.3103	0.2671
0.1013	22.22	2400	0.2589	0.2324
0.0704	25.92	2800	0.2826	0.2339
0.0537	29.63	3200	0.2704	**0.2310**

Şekil 16: Modelin Performans Skorları

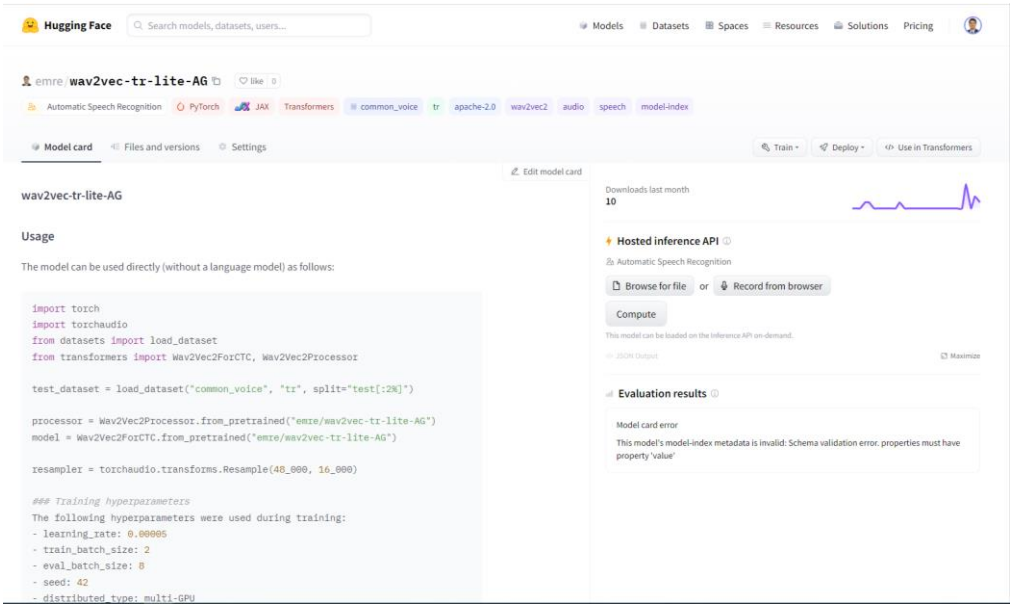


Kaynak: Yazar tarafından derlenmiştir.

3.4. MODELİN AÇIK KAYNAKLI PAYLAŞIMI

Eğitimi gerçekleştirilen bu model açık kaynaklı bir yapay zekâ paylaşım platformu olan huggingface.co isimli web adresinde yazar tarafından oluşturulmuş olan, emre/wav2vec-tr-lite-AG isimli model olarak paylaşıma sunulmuştur. Paylaşılan modelin ekran görüntüsü Şekil 17'de gösterilmektedir. Buradan yapılan kontrollerin doğrulanması için tahminleme fonksyonuna her bir harf ve boşluğa [PAD] harf veya boşluk [PAD] şeklinde ayrım yaparak tahminleme gerçekleştirecek bir arayüz üzerinden yapılacak sorgular test edilmiştir. Modelin test için ayrılan veri seti ile alınan sonuçlara göre kelime WER 0,23, CER 0,1 olarak bulunmuştur. Bireysel olarak yapılan tekil sorgularda da bu oranlar ile korele sonuçlar alınmış, modelin performansı 2 ayrı şekilde test edilmiştir. Gerçekleştiren bu model eğitimi için gerekli olan GPU gereksinimi yüksek olduğu için ayrıca Google Colab üzerinden modelin eğitildiği kodlar kullanılarak daha distile bir model üretebilmek için bir jupyter not defteri paylaşılmıştır. Bu not defteri kullanılarak Türkçe dilinde veya farklı dillerde dil modelleri eğitmek mümkün olmakla birlikte kullanacak kişiler için detaylı açıklamalar da eklenmiştir.

Şekil 17: Yazar Tarafından Açık Kaynaklı Paylaşılan Model



Kaynak: Yazar tarafından derlenmiştir.

SONUÇ

Bu projede Türkçe dili için az miktarda veriye dayalı ASR sistemi uygulanmıştır. SSL, bu araştırmayı yürütmek için kullanılan ana araç olmuştur. İlk olarak, literatürde yapılmış benzer ASR modelleri incelenmiş ve bu önceden eğitilmiş modellerin etiketli Türkçe veriler üzerinde performansları incelenmiştir. Elde edilen sonuçların daha da iyileştirilmesi için bu çalışma gerçekleştirilmiştir. Daha sonra ise gerçekleştirilen bu çalışma ile bir çağrı merkezinde kullanım senaryosu oluşturularak, organizasyonel gelişim için bir otomatik konuşma tanıma sistemi kurgusu yapılmıştır.

Bu çalışmada, Türkçe de dahil olmak üzere birçok dilde önceden eğitilmiş denetimsiz bir XLSR'den yararlanılmıştır. Ardından bu önceden eğitilmiş model, Türkçe dilinde öğrenme aktarması yolu ile ince ayar yapılarak kullanılabilecek hale getirilmiştir. Denetimsiz XLSR, 53 farklı dilde önceden eğitilmiş ve maskeli gizli konuşma temsilleri için karşıt bir görevi çözen Wav2Vec 2.0 üzerine inşa edilmiştir. Türkçe dilinde Wav2Vec 2.0 ile ince ayar yapılmış XLSR'nin WER değeri başlangıçta 0,97 idi. Model parametrelerinde ince ayar yaptıktan sonra, bu modeli hugingface şirketinin Türkçe dil modellerinde üst sıralara yerleştiren 0,23'lük bir WER elde edilmiştir. Şu anda üretmiş olan bu model aktif olarak kullanılıyor ve hugingface'in resmî web sitesinden 60'tan fazla kez indirilmiştir.

Toplulukların farklı diller için konuşma külliyatı oluşturmaya katılmalarını sağlayan Mozilla Common Voice gibi açık kaynaklı projeler, yazıya dökülmüş verilerin yetersiz olması problemine bir çözüm üretmek için ortaya çıkmıştır. Bu çalışmada kullanılan konuşma verileri Mozilla Common Voice'tan elde edilmiştir. Ek olarak, büyük miktarda etiketli veri bulmak, burada kullanılan öğrenme aktarması yöntemleri ile önemsiz bir problem haline gelmiştir. Bu araştırmanın konusu olan denetimsiz öğrenme yönteminin, etiketlenmemiş Türkçe konuşma verileriyle uğraşırken başarılı olduğu görülmüştür. Bu araştırmada, eğitim öncesi aşamada birden fazla dilden etiketlenmemiş verileri kullanmak ve ardından modelin etiketli Türkçe dil verileri üzerinde ince ayar yapmak için bu araştırmada XLSR kullanılmıştır. ASR için önceden eğitilmiş modelleri kullanmak ve bunları etiketli Türkçe konuşma verilerinde ince ayar yapmak, tek dilli bir modeli önceden eğitmekten daha verimli ve zaman

kazandıran bir yöntemdir. Ayrıca çalışmada, 30 saat kadar az etiketli veri kullanarak 0,23'lük bir WER elde edilebileceği gösterilmiştir.

Etiketlenmiş veri ile Wav2Vec2 modelleri, geleneksel ASR sistemlerinden daha iyi performans gösterebilir. Ayrıca, ham bir sesi uçtan uca bir şekilde bir dizi grafik veya kelimeye eşlerler, böylece ayrı bileşenleri eğitmeye gerek kalmaz. Bununla birlikte, bu modellerin oldukça büyük olduğu yüz milyonlarca parametreden oluştuğu, gerçek zamanlı olarak konuşmadan metne dönüştürme olasılığını ortadan kaldırdığı ve kabul edilebilir kod çözme hızı için bir GPU gibi verimli bir donanım gerektirdiğine dikkat edilmelidir. Bu nedenle gelecekteki araştırmalar, model boyutunu ve model parametrelerinin sayısını azaltmak ve ayrıca çıkarımı hızlandırmak için Wav2Vec2 modellerine sıkıştırma tekniklerinin uygulanmasıyla ilgili deneyleri içerebilir.

Sonuç olarak, bu yüksek lisans tezi, kurumsal gelişimi iyileştirmek için çağrı merkezlerinde kullanılmak üzere otomatik bir konuşma tanıma sistemi önermiştir. Önerilen sistemin entegrasyonu ve performansa olan etkisini ölçümlemek için bir çağrı merkezinde yapılacak kontrollü deney senaryosu oluşturulmuştur. Sistem, test edilmiş ve etkili olduğu gösterilmiş teorik olarak tanımlanmış bir mimariye dayanmaktadır. Sistemin bir çağrı merkezinde kullanılmasının kurumun verimliliğini ve müşteri memnuniyetini artırması beklenmektedir. Bu çalışmanın sonuçları, önerilen otomatik konuşma tanıma sisteminin kurumsal bir ortamda uygulanmasının potansiyel faydalarını göstermektedir.

Ayrıca, bu sistemin kullanımı bir çağrı merkezinin genel operasyonlarını iyileştirme potansiyeline sahiptir. Sistem, belirli görevleri ve süreçleri otomatikleştirerek, çağrı merkezi çalışanlarının iş yükünün azaltılmasına yardımcı olabilir ve daha karmaşık ve katma değerli görevlere odaklanmalarını sağlar. Bu, organizasyonda üretkenlikte ve genel verimlilikte bir artışa yol açabilir.

Ek olarak, önerilen sistem müşteri deneyimini iyileştirmek için tasarlanmıştır. ASR teknolojisini kullanan sistem, müşteri sorularına hızlı ve doğru yanıtlar vererek müşteri için daha verimli ve tatmin edici bir deneyim sağlayabilir. Bu, müşteri memnuniyetini ve sadakatini artırmaya yardımcı olarak kuruluş için artan gelire yol açabilir.

Özetle, önerilen bu ASR sistemi, bir çağrı merkezi ortamında kurumsal gelişimi önemli ölçüde geliştirme potansiyeline sahiptir. Sistem, belirli görevleri ve

süreçleri otomatikleştirerek üretkenliği ve verimliliği artırmaya yardımcı olabilir, bu da gelişmiş operasyonlara ve organizasyonel performans artışına yol açar.

Önerilen ASR sisteminin uygulanmasını daha fazla desteklemek için, sistemin sağlayabileceği potansiyel maliyet tasarruflarını dikkate almak önemlidir. Sistem, belirli görevleri ve süreçleri otomatikleştirerek insan emeğine olan ihtiyacın azaltılmasına yardımcı olabilir ve bu da kuruluş için önemli maliyet tasarrufları sağlayabilir. Bu, özellikle yüksek hacimli çağrıları işlemek için çok sayıda çalışana ihtiyaç duyan çağrı merkezleri için faydalı olabilir.

Geliştirmiş olduğumuz otomatik konuşma tanıma sisteminin konuşmadan metne dönüştürme işlemlerini, tespit edilmiş olan kelime hata oranı ile gerçekleştirdiğini, bunu yaparken eğitilmiş olduğu veri seti kısıtları içerisinde (yaş, cinsiyet) bunu gerçekleştirebildiğini gördük. Çağrı merkezleri özelinde, çağrı merkezine ait konuşma veri setleri ile karşılık gelen metinler kullanılarak, bu mimarideki yapay zekâ modelinin tekrar eğitilmesi durumunda, amacına uygun olarak daha yüksek bir performans gösterebilecek bir modelin ortaya çıkartılabilme imkânı bulunmaktadır. Elde edilen bu model ile kullanılabilecek olan yazılım mimarisini tasarlarken, farklı yazılımlar ile kolaylıkla entegre olabilmesi için bir servis düzeyindeki bir mimari tasarım önerilmiştir. Yapay zekâ modelinin önerilen entegrasyon ile kullanılması halinde yukarıda bahsedilen faydaları sağlayacağı görülmüştür. Bu nedenle de bu çalışma ile önerilen otomatik konuşma tanıma sisteminin, önerilen şekilde kullanılması durumunda çağrı merkezi örneğinde organizasyonel gelişime olumlu yönde katkıda bulunacağı sonucuna ulaşılmıştır.

KAYNAKÇA

Abdel-Hamid, O., Mohamed, A. R., Jiang, H., Deng, L., Penn, G., & Yu, D. (2014). Convolutional neural networks for speech recognition. *IEEE/ACM Transactions on audio, speech, and language processing*, 22(10), 1533-1545.

Amodei, D., Ananthanarayanan, S., Anubhai, R., Bai, J., Battenberg, E., Case, C., ... & Zhu, Z. (2016, June). Deep speech 2: End-to-end speech recognition in english and mandarin. In *International conference on machine learning* (pp. 173-182). PMLR.

Ba, J. L., Kiros, J. R., & Hinton, G. E. (2016). Layer normalization. *arXiv preprint* arXiv:1607.06450.

Baevski, A., Zhou, Y., Mohamed, A., & Auli, M. (2020). wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in Neural Information Processing Systems*, 33, 12449-12460.

Baevski, A., Schneider, S., & Auli, M. (2019). vq-wav2vec: Self-supervised learning of discrete speech representations. *arXiv preprint* arXiv:1910.05453.

Chi, P. H., Chung, P. H., Wu, T. H., Hsieh, C. C., Chen, Y. H., Li, S. W., & Lee, H. Y. (2021, January). Audio albert: A lite bert for self-supervised learning of audio representation. In *2021 IEEE Spoken Language Technology Workshop (SLT)* (pp. 344-350). IEEE.

Chiu, C. C., Sainath, T. N., Wu, Y., Prabhavalkar, R., Nguyen, P., Chen, Z., ... & Bacchiani, M. (2018, April). State-of-the-art speech recognition with sequence-to-sequence models. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 4774-4778). IEEE.

Çetin, D. (2011). İhtiyaç Belirleme Teknikleri. http://dursuncetin.blogspot.com.tr/2011/04/ihtiyac-belirleme-teknikleri.html,
(18.04.2017).

Danis, C., & Karat, J. (1995, August). Technology-driven design of speech recognition systems. In *Proceedings of the 1st conference on Designing interactive systems: processes, practices, methods, & techniques* (pp. 17-24).

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint* arXiv:1810.04805.

Feng, W., Guan, N., Li, Y., Zhang, X., & Luo, Z. (2017, May). Audio visual speech recognition with multimodal recurrent neural networks. In 2017 *International Joint Conference on Neural Networks* (IJCNN) (pp. 681-688). IEEE.

Filippidou, F., & Moussiades, L. (2020, June). A benchmarking of IBM, Google and Wit automatic speech recognition systems. In *IFIP International Conference on Artificial Intelligence Applications and Innovations* (pp. 73-82). Springer, Cham.

Fukushima, K. (1980). Neocognitron: A self-or~anizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*. 36(4), 193-202.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Convolutional networks. In *Deep learning* (Vol. 2016, pp. 330-372). Cambridge, MA, USA: MIT Press.

Graves, A., Fernández, S., Gomez, F., & Schmidhuber, J. (2006, June). Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning* (pp. 369-376).

Heming, Z., Xudong, Z., & Yibiao, Y. (2000, August). Co-channel speech separation based on sinusoidal model for speech. In WCC 2000-ICSP 2000. 2000 *5th International Conference on Signal Processing Proceedings. 16th World Computer Congress 2000* (Vol. 2, pp. 815-818). IEEE.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8), 1735-1780.

Kabir, M. M., Mridha, M. F., Shin, J., Jahan, I., & Ohi, A. Q. (2021). A survey of speaker recognition: Fundamental theories, recognition methods and opportunities. *IEEE Access*, 9, 79236-79263.

Kalchbrenner, N., Grefenstette, E., & Blunsom, P. (2014). A convolutional neural network for modelling sentences. *arXiv preprint* arXiv:1404.2188

Kan, S. H. (2003). *Metrics and models in software quality engineering*. Addison-Wesley Professional.

Laudon, K.C., Laudon, J. P. (2014). *Management Information Systems (12th Edition)*. ISBN-13: 978-0132142854.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278-2324.

Liu, A. T., Yang, S. W., Chi, P. H., Hsu, P. C., & Lee, H. Y. (2020, May). Mockingjay: Unsupervised speech representation learning with deep bidirectional transformer encoders. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing* (ICASSP) (pp. 6419-6423). IEEE.

Matsumoto, T., Yokohama, T., Suzuki, H., Furukawa, R., Oshimoto, A., Shimmi, T., ... & Chua, L. O. (1990, December). Several image processing examples by CNN. In IEEE *International Workshop on Cellular Neural Networks and their Applications* (pp. 100-111). IEEE.

McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4), 115-133.

Mohamed, A., Okhonko, D., & Zettlemoyer, L. (2019). Transformers with convolutional context for ASR. *arXiv preprint* arXiv:1904.11660.

Olah, C. (2015). Understanding lstm networks.

Oyucu, S., Polat, H., & Sever, H. (2020). Sessizliğin kaldırılması ve konuşmanın parçalara ayrılması işleminin Türkçe otomatik konuşma tanıma üzerindeki etkisi. *Düzce Üniversitesi Bilim ve Teknoloji Dergisi*, 8(1), 334-346.

Pallett, D. S. (2003, November). A look at NIST's benchmark ASR tests: past, present, and future. In 2003 *IEEE Workshop on Automatic Speech Recognition and Understanding* (IEEE Cat. No. 03EX721) (pp. 483-488). IEEE.

Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., ... & Vesely, K. (2011). The Kaldi speech recognition toolkit. In *IEEE 2011 workshop on automatic speech recognition and understanding* (No. CONF). IEEE Signal Processing Society.

Rashmi, C. R. (2014). Review of algorithms and applications in speech recognition system. *International Journal of Computer Science and Information Technologies*, 5(4), 5258-5262.

Prabhavalkar, R., Rao, K., Sainath, T. N., Li, B., Johnson, L., & Jaitly, N. (2017, August). A Comparison of Sequence-to-Sequence Models for Speech Recognition. In *Interspeech* (pp. 939-943).

Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6), 386.

Sehgal, R. R., & Raj, G. (2018, April). Interactive Voice Response Using Automatic Speech Recognition Techniques for Call Centers. In *Proceedings of 3rd International Conference on Internet of Things and Connected Technologies* (ICIoTCT) (pp. 26-27).

Song, Y., Lian, R., Chen, Y., Jiang, D., Zhao, X., Tan, C., ... & Wong, R. C. W. (2022, October). A Platform for Deploying the TFE Ecosystem of Automatic Speech Recognition. In *Proceedings of the 30th ACM International Conference on Multimedia* (pp. 6952-6954).

Van Den Oord, A., & Vinyals, O. (2017). Neural discrete representation learning. *Advances in neural information processing systems*, 30.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems* (pp. 5998-6008).

Wu, J. (2017). Introduction to convolutional neural networks. *National Key Lab for Novel Software Technology. Nanjing University. China*, 5(23), 495.

Xiong, W., Wu, L., Alleva, F., Droppo, J., Huang, X., & Stolcke, A. (2018, April). The Microsoft 2017 conversational speech recognition system. In *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (pp. 5934-5938). IEEE.

Yakar, Ö., & Aşlıyan, R. (2016). Saklı Markov Modeli Kullanarak Türkçe Konuşma Tanıma. *Akademik Bilişim*, 16, 1-7.

Yamashita, R., Nishio, M., Do, R. K. G., & Togashi, K. (2018). Convolutional neural networks: an overview and application in radiology. *Insights into imaging*, 9(4), 611-629.

Yao, L., Mao, C., & Luo, Y. (2019, July). Graph convolutional networks for text classification. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 33, No. 01, pp. 7370-7377).

Yi, L. I. N., Min, R. U. A. N., Kunjie, C. A. I., Dan, L. I., Ziqiang, Z. E. N. G., Fan, L. I., & Bo, Y. A. N. G. (2022). Identifying and managing risks of AI-driven operations: A case study of automatic speech recognition for improving air traffic safety. *Chinese Journal of Aeronautics*.

Yu, D., & Deng, L. (2016). *Automatic speech recognition (Vol. 1)*. Berlin: Springer.

Zekveld, A. A., Kramer, S. E., Kessens, J. M., Vlaming, M. S., & Houtgast, T. (2009). The influence of age, hearing, and working memory on the speech comprehension benefit derived from an automatic speech recognition system. *Ear and hearing*, 30(2), 262-272.

Zhao, Y., Yin, D., Luo, C., Zhao, Z., Tang, C., Zeng, W., & Zha, Z. J. (2021). General-Purpose Speech Representation Learning through a Self-Supervised Multi-Granularity Framework. *arXiv preprint arXiv*:2102.01930.

Zweig, G., Siohan, O., Saon, G., Ramabhadran, B., Povey, D., Mangu, L., & Kingsbury, B. (2006, May). Automated quality monitoring in the call center with asr and maximum entropy. In *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings* (Vol. 1, pp. I-I). IEEE.

Tecim, V. (2004) Sistem Geliştirme Yaşam Döngüsü. *http://debis.deu.edu.tr/ userweb// vahap.tecim/dosyalar/sgyd.pdf.* (15.04.2017).

Mozilla Common Voice Web Sitesi (2021). *DATASET* (https://discourse.mozilla.org/t/common-voice-turkcenin-durumu/85895, (10.05.2022)

Taşar D.E. (2021). Çalışmaya ait not defteri - *Google Collabotory Platform*. (https://colab.research.google.com/drive/1xcfnBFdtT6HcW6MDLgImj7tJcYafJSJU?usp=sharing), (01.12.2022)

Taşar D.E. (2021). Geliştirilen yapay zeka modelinin açık kaynaklı paylaşım linki - *Huggingface Platform*, (https://huggingface.co/emre/wav2vec-tr-lite-AG), (10.05.2022)