UÇTAN-UCA KONUŞMA TANIMA MODELİ: TÜRKÇE'DEKİ DENEYLER

END-TO-END SPEECH RECOGNITION MODEL: EXPERIMENTS IN TURKISH

BEHNAM ASEFISARAY

Prof.Dr. HAYRİ SEVER
Tez Danışmanı

Yrd.Doç.Dr. ERHAN MENGÜŞOĞLU
İkinci Tez Danışmanı

Hacettepe Üniversitesi
Lisansüstü Eğitim-Öğretim ve Sınav Yönetmenliğinin
Bilgisayar Mühendisliği Anabilim Dalı için Öngördüğü
DOKTORA TEZİ olarak hazırlanmıştır

2018

BEHNAM ASEFISARAY'ın hazırladığı "**Uçtan-Uca Konuşma Tanıma Modeli: Türkçe'deki Deneyler**" adlı bu çalışma aşağıdaki jüri üyeleri tarafından BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI'nda DOKTORA TEZİ olarak kabul edilmiştir.

Prof.Dr. Mehmet Önder EFE
Başkan

Prof.Dr. Hayri SEVER
Danışman

Yrd.Doç.Dr. Mustafa SERT
Üye

Yrd.Doç.Dr. Gönenç ERCAN
Üye

Yrd.Doç.Dr. Abdül Kadir GÖRÜR
Üye

Bu tez Hacettepe Üniversitesi Fen Bilimleri Enstitüsü tarafından DOKTORA TEZİ olarak onaylanmıştır.

Prof.Dr. Menemşe GÜMÜŞDERELİOĞLU
Fen Bilimleri Enstitüsü Müdürü

YAYINLAMA VE FİKRİ MÜLKİYET HAKLARI BEYANI

Enstitü tarafından onaylanan lisansüstü tezimin/raporumun tamamını veya herhangi bir kısmını, basılı (kağıt) ve elektronik formatta arşivleme ve aşağıda verilen koşullarla kullanıma açma iznini Hacettepe üniversitesine verdiğimi bildiririm. Bu izinle Üniversiteye verilen kullanım hakları dışındaki tüm fikri mülkiyet haklarım bende kalacak, tezimin tamamının ya da bir bölümünün gelecekteki çalışmalarda (makale, kitap, lisans ve patent vb.) kullanım hakları bana ait olacaktır.

Tezin kendi orijinal çalışmam olduğunu, başkalarının haklarını ihlal etmediğimi ve tezimin tek yetkili sahibi olduğumu beyan ve taahhüt ederim. Tezimde yer alan telif hakkı bulunan ve sahiplerinden yazılı izin alınarak kullanması zorunlu metinlerin yazılı izin alarak kullandığımı ve istenildiğinde suretlerini Üniversiteye teslim etmeyi taahhüt ederim.

☐ **Tezimin/Raporumun tamamı dünya çapında erişime açılabilir ve bir kısmı veya tamamının fotokopisi alınabilir.**
(Bu seçenekle teziniz arama motorlarında indekslenebilecek, daha sonra tezinizin erişim statüsünün değiştirilmesini talep etseniz ve kütüphane bu talebinizi yerine getirse bile, tezinin arama motorlarının önbelleklerinde kalmaya devam edebilecektir.)

✓ **Tezimin/Raporumun 12/01/2019 tarihine kadar erişime açılmasını ve fotokopi alınmasını (İç Kapak, Özet, İçindekiler ve Kaynakça hariç) istemiyorum.**
(Bu sürenin sonunda uzatma için başvuruda bulunmadığım taktirde, tezimin/raporumun tamamı her yerden erişime açılabilir, kaynak gösterilmek şartıyla bir kısmı ve ya tamamının fotokopisi alınabilir)

☐ **Tezimin/Raporumun ..... tarihine kadar erişime açılmasını istemiyorum, ancak kaynak gösterilmek şartıyla bir kısmı veya tamamının fotokopisinin alınmasını onaylıyorum.**

☐ **Serbest Seçenek/Yazarın Seçimi**

15 /01 /2018

BEHNAM ASEFISARAY

Aileme…

ETİK

Hacettepe Üniversitesi Fen Bilimleri Enstitüsü, tez yazım kurallarına uygun olarak hazırladığım bu tez çalışmasında,

- tez içindeki bütün bilgi ve belgeleri akademik kurallar çerçevesinde elde ettiğimi,
- görsel, işitsel ve yazılı tüm bilgi ve sonuçları bilimsel ahlak kurallarına uygun olarak sunduğumu,
- başkalarının eserlerinden yararlanılması durumunda ilgili esere bilimsel normlara uygun olarak atıfta bulunduğumu,
- atıfta bulunduğum eserlerin tümünü kaynak olarak gösterdiğimi,
- kullanılan verilerde herhangi bir tahrifat yapmadığımı,
- ve bu tezin herhangi bir bölümünü bu üniversite veya başka üniversitede başka bir tez çalışması olarak sunmadığımı

beyan ederim.

15/01/2018

BEHNAM ASEFISARAY

ÖZET

UÇTAN-UCA KONUŞMA TANIMA MODELİ: TÜRKÇE'DEKİ DENEYLER

Behnam ASEFISARAY

Doktora, Bilgisayar Mühendisliği Bölümü

Tez Danışmanı: Prof.Dr. Hayri SEVER

İkinci Tez Danışmanı: Yrd.Doç.Dr. Erhan MENGÜŞOĞLU

Ocak 2018, 125 sayfa

Okunuş sözlüğü ve saklı markov modeli (Hidden Markov Model - HMM) yıllardır konuşma tanıma sistemlerinin en önemli iki parçası olarak bilinmekteler. HMM'ler çıktı olarak ürettikleri fonemler arasında bağımsızlık varsayımında bulunup, sözlükteki kelimelerin okunuşunu el yordamı ile oluşturmak da oldukça zaman alıcı bir süreçtir. Ayrıca bu modellerin eğitimi de birbirinden bağımsız yapılıp, bir modeldeki iyileşme her zaman konuşma tanıma sisteminin hata oranını düşürmemektedir. Son yıllarda, bağlantıcı zamansal sınıflandırma (Connectionist Temporal Classification - CTC) yöntemi bu sorunu kısmen çözmüş olup akustik model ile okunuş modelinin birlikte eğitilebilmesini sağlamıştır. Ancak hem HMM hem de CTC çözümleri, karakter/kelime çıktıları arasında bağımsızlık varsayımında bulunup, gerek akustik gerekse okunuş açısından uzun bağımlılıkları modelleyememekteler. Bu nedenden dolayı da, HMM ve CTC tabanlı sistemler her zaman güçlü bir dil modeline ihtiyaç duyup, dil modeli kullanmadan bu sistemlerdeki kelime hata oranı oldukça yüksek çıkmaktadır.

Bu tezde, HMM tabanlı sistemlerin yapısı incelenip bu modellerin getirdiği kısıtlamalar anlatılmıştır. Odaklanma mekanizması (Attention Mechanism) ile çalışan bir tekrarlanan sinir ağı (Recurrent Neural Network - RNN) direkt sesi yazıya çevirmek için eğitilip, yukarıdaki kısıtlamalar ve bağımsızlıklar olmadan Türkçe konuşma tanıma sisteminin yapısı verilmiştir. Kullanılan bu model, uçtan uca eğitilip konuşma tanıma sisteminin içerisinde bulunması gereken okunuş sözlüğü, dil modeli ve akustik model tek bir model kapsamında eğitilmiştir. Bu sayede, farklı modellerin birbirinden bağımsız olarak eğitilmesine gerek kalmayıp nihai sonucu iyileştirecek ve bütün bağımlılıkları göz önünde bulundurabilecek bir model tasarımı ve eğitimi yapılmıştır. Transfer öğrenme yöntemi kullanarak uçtan uca bir konuşma tanıma modeli daha az veriyle eğitilip yeterince iyi bir model elde edilmiştir.

Anahtar Kelimeler: konuşma tanıma sistemi, akustik model, dil modeli, okunuş sözlüğü, saklı markov modeli, bağlantıcı zamansal sınıflandırma, tekrarlanan sinir ağı, odaklanma mekanizması.

ABSTRACT

END-TO-END SPEECH RECOGNITION MODEL: EXPERIMENTS IN TURKISH

Behnam ASEFISARAY

Doctor of Philosophy, Department of Computer Engineering

Supervisor: Prof.Dr. Hayri SEVER

Co-Supervisor: Asst. Prof.Dr. Erhan MENGÜŞOĞLU

January 2018, 125 pages

For decades, the main components of Automatic Speech Recognition (ASR) systems have been pronunciation dictionary and Hidden Markov Models (HMMs). HMMs assume conditional independence between its output and creating the pronunciation dictionary have a tedious and time consuming process. Additionally, training each of these models are independent with each other and there especially exists a disconnect between acoustic model accuracy and word error rate (Word Error Rate) of automatic speech recognition. Connectionist Temporal Classification (CTC) character models attempts to solve some of these issues by jointly learning the pronunciation and acoustic model as a single model. However, both HMM and CTC models suffer from conditional independence assumption and rely heavily on a large enough language model during decoding.

In this thesis, we investigate the traditional paradigm of ASR and focus the limitations of HMM and CTC base speech recognition models. We propose an approach to ASR with neural attention mechanism models and we directly optimize speech transcriptions error rate in Turkish. The end-to-end recurrent neural network model jointly learns all the main components of a speech recognition system: the pronunciation dictionary, language model and acoustic model. We used transfer learning in our end-to-end architecture in order to training a good enough acoustic model using limited amount of transcribed speech data.

Keywords: Speech recognition, acoustic model, language model, pronunciation dictionary, hidden markov model, connectionist temporal classification, recurrent neural network, attention mechanism.

TEŞEKKÜR

Tez çalışmalarım süresince engin bilgi birikiminin yanı sıra gerek akademik gerekse hayata dair tecrübelerini benimle paylaşarak bu süreçte bana hoşgörü ve sabırla destek olmanın ötesinde bilhassa doktora sürecimin en zor dönemlerinde ışığıyla beni ve yolumu aydınlatan danışmanım Sayın Prof.Dr. Hayri SEVER'e,

Doktora sürecimin ilk dönemlerinden itibaren bu alandaki çalışmalarımda büyük bir özveri ile her daim bana yol gösteren ve sürekli beni motive eden eş danışmanım Sayın Yrd.Doç.Dr. Erhan MENGÜŞOĞLU'na,

Çok değerli yorum ve önerileriyle tez çalışmalarıma değerli katkılarını sunan tez savunma sınavı jüri üyelerim Sayın Prof.Dr. Mehmet Önder EFE'ye, Sayın Yrd.Doç.Dr. Mustafa SERT'e, Sayın Yrd.Doç.Dr. Gönenç ERCAN'a ve Sayın Yrd.Doç.Dr. Abdül Kadir Görür'e,

Eğitimim süresince hep gurur duyduğum ve kendimi bir parçası olarak gördüğüm Hacettepe Üniversitesi Bilgisayar Mühendisliği Bölümü'nün tüm akademik ve idari çalışanlarına,

Bu sürecin son anına kadar bana her anlamda destek olan ve sürekli fikir alışverişinde bulunduğum iş arkadaşım, mesai arkadaşım ve müdürüm Sayın Ali HAZNEDAROĞLU başta olmak üzere desteklerini esirgemeyen Sayın Anıl ÖZTUNCER'e, Sayın Kaan KAYA'ya ve Sayın Murat İLKDOĞAN'a,

Bu süreçte ve hep yanımda olan, beni destekleyen, büyük sevgi ve aşkını hiçbir zaman benden esirgemeyen hayat arkadaşım, zorlu günlerimin dostu eşim *Mana*'ya,

Hayata gözlerimi açtığım ilk andan itibaren hiçbir fedakârlıktan kaçınmayarak sevgi, destek ve ilgileriyle bugünümün ve yarınlarımın baş mimarları olan anneme ve babama sonsuz teşekkür, sevgi ve saygılarımı sunarım.

İÇİNDEKİLER

Sayfa

ÇİZELGELER

Sayfa

ŞEKİLLER

Sayfa

SİMGELER VE KISALTMALAR

Simgeler

p_{AM}	Akustik Modelin ürettiği olasılık değeri
p_{LM}	Dil Modelin ürettiği olasılık değeri
p(W)	Kelime sırasının olasılığı
p(W|X)	X Sinyalinin W Kelimesini Üretme Olasılığı
β	Dil modelinin ağırlığı
λ	Modelin Eğitiminin Öğrenme Oranı
Ɵ	Sinir Ağları Modelinin Parametreleri
∂	Zincir Türevler

Kısaltmalar

KTS	Konuşma Tanıma Sistemi
SMM	Saklı Markov Modeli
SA	Sinir Ağları
TSA	Tekrarlanan Sinir Ağları
İBSA	İleri Beslemeli Sinir Ağı
AM	Akustik Model
DM	Dil Modeli
DSA	Derin Sinir Ağı
DOY	Dinle Odaklan Yaz

1. GİRİŞ

1.1. Tez Çalışmasının Motivasyonu

İnsanların, bilgisayar ve diğer elektronik cihazlar ile iletişim kurmalarının en eski yöntemlerinden birisi klavye kullanmaktır. Fiziksel klavyeleri kullanarak bilgisayarlara komut gönderip bunun karşılığında bir cevap alıyoruz. Örneğin, cep telefonu kullanmanın tek yolu birkaç sene öncesine kadar sadece telefonlar üzerindeki klavyelerdi. Telefon üzerindeki bu tuşları kullanarak, insanlar mesaj yazma ve numara çevirme gibi işlemleri bu cihazlar üzerinde yapmaktalar. Daha sonra, fare (mouse) gibi diğer girdi cihazlarının icadı ile birlikte ekrandaki objeler üzerinde gezinerek işlem yapmak klavyeye göre oldukça zaman kazandırdı. Son yıllarda, dokunmatik ekranların piyasaya sürülmesi ile birlikte, ekrana dokunmak akıllı cihazlarla iletişime geçmenin en temel ve hatta birçok cihaz için tek seçenek oldu. Akıllı telefonlar, bilgisayarlar, beyaz eşya ve daha birçok elektronik cihazların ara yüzünde bu dokunmatik ekranlar bulunup bilgisayar ile iletişim kurmayı daha da zevkli ve hızlı bir hale getirdi.

Her geçen gün daha çabuk bir şekilde günlük işlerimizi yapıp daha fazla zaman kazanmak istiyoruz. Bu yüzden de, iletişim kurmamız gereken varlıklar ile hızlı ve kolay bir şekilde iletişime geçebilmemiz bize hem zaman kazandırıp hem de hayatı kolaylaştıracaktır. İnsanlar birbirleri ile konuşarak çok daha hızlı anlaşıp ve istedikleri maksimum bilgiyi minimum sürede aktarabiliyorlar. Diğer bir taraftan da yazılı olarak bir insan ile iletişime geçmek hem zaman açısından hem de aktarılması gereken duygular açısından konuşmaya göre daha zor ve zaman alıcı bir yöntemdir. İnsan, bu alışkanlığı ve yeteneği yüzünden hep konuşarak bilgisayar ile iletişim kurmayı hayal etmiştir. Dolayısıyla, klavye kullanmak yerine konuşarak bir bilgisayara komut vermek bizim için hem daha kolay olup hem de karşımızdaki bilgisayara canlı bir varlık olarak bakmamızı sağlayacaktır.

Konuşma sinyalinin bir bilgisayar tarafından tanınması bilgisayarlarla iletişim kurabilmenin en temel adımlarından birisidir. Kelimelerin bilgisayar tarafından algılanması (detection) ve tanınması (recognition) sonucunda, bu varlıklarla daha doğal bir şekilde iletişime geçip karşımızdaki cihazın canlı olduğunu hayal etmemiz biraz daha kolaylaşacaktır. Konuşma tanıma sisteminin (KTS) temel amacı, ses

sinyallerinin bilgisayar tarafından algılanıp örüntü tanıma algoritmaları (pattern recognition) ve modellerini kullanarak bu sinyali yazıya veya farklı bir bilgi şekline dönüştürmektir. Konuşmayı yazıya çevirmek, farklı bir ses sentezi üretmek, duygu analizi (sentiment analysis) ve konuşma anlama (speech understanding) gibi işlevler ancak başarılı bir konuşma tanıma sistemi ve modelin ortaya konulması ile mümkün olacaktır. Dolayısıyla, bütün bu doğal iletişim kanallarının iyileşmesi ve hata oranının düşürülmesindeki en önemli etkenlerden birisi olan konuşma tanımanın farklı diller için gelişmesi ve iyileşmesi gerekmektedir.

Konuşma tanıma teknolojisini sesi metine çevirmek için kullanmak, bu sistemlerin en çok kullanıldığı alanlardan birisidir. Bu teknoloji dikte yapma, komut kontrol sistemleri, akıllı ev asistanı, doğal dil işleme, makine çevirisi ve çağrı merkezlerinde konuşma analizi (speech analytics) gibi çeşitli uygulamalarda kullanılabilir. Tüm bu uygulamaların ve ürünlerin özünde konuşma tanıma teknolojisi yer alıp bu teknolojinin gelişmesi ve kelime hata oranındaki düşüş ilgili ürünlerin kalitesini doğrudan etkilemektedir.

Konuşma tanıma teknolojisinin kullanımı son birkaç sene içerisinde çok daha yaygınlaşmaya başladı. Bu sistemlerdeki doğruluk ve güvenirlilik seviyesindeki artış ile birlikte insanlar bu sistemlere daha çok güvenmeye başlayıp günlük işlemlerini artık cihazlara konuşarak gerçekleştiriyorlar. Daha karmaşık ve güçlü modellerin kullanımı bir taraftan, hızlı ve yüksek kapasiteli işlemci ve bellekler de diğer taraftan bu teknolojinin çok daha büyük veri kümeleri ve karmaşık modeller üzerine inşa edilip başarısının oldukça artmasına yol açmıştır. Örneğin, milyonlarca parametre içeren derin sinir ağları (deep neural network - DNN) tabanlı bir model binlerce saat eğitim verisi ile grafik işleme birimleri (graphical processing unit - GPU) donanımı sayesinde birkaç gün içerisinde eğitilebiliyor [1].

Dünyada konuşulan birçok doğal dil bulunup konuşma tanıma sistemlerinin doğruluk oranı tüm bu diller için aynı seviyede değildir. Bunun sebebi de bu teknolojide kullanılan yöntemler ve algoritmaların istatistiksel makine öğrenimi (statistical machine learning) kavramlarına dayalı olmasından kaynaklanmaktadır. Yeni bir dil için konuşma tanıma teknoloji desteği sağlamak için o dile ait sesli ve yazılı veri örneklerinin elde dilip istatistiksel modellerin eğitilmesi gerekmektedir. Fakat bu süreç oldukça pahalı (zaman ve kaynak açısından) bir süreç olduğundan

dolayı daha az çalışılmış diller için konuşma tanımanın doğruluk oranı düşük kalıp bu diller için yeterince eğitim verisi toplamak gerekmektedir [2, 3].

Gelişkin (state-of-the-art) konuşma tanıma sistemleri oldukça kompleks bir yapıya sahip olup farklı bileşenlerin birlikte çalışması sonucunda tanıma gerçekleştiriliyor. Okunuş modeli (pronunciation model), akustik model (acoustic model), dil modeli (language model) ve metin düzenleme (text normalization) gibi farklı modüllerin tasarlanıp çoğu zaman el yordamı ile ayarlanması gerekmektedir [1, 4, 5]. Bu süreçteki aşamaların çoğu da insanın direkt müdahalesi ile tamamlanıp farklı durumlar için farklı parametre değerleri kullanılmaktadır (deşifre aşamasındaki dil modelinin ağırlığı gibi) [1, 4, 5].

Konuşma tanıma sistemindeki her bir modül, modellemeye çalıştığı veri kümesi ve yapısı konusunda kendi içerisinde birtakım varsayımlarda bulunuyor [1, 5]. Ayrıca bu modeller genelde tek başlarına eğitilip konuşma tanıma sisteminde kullanılan diğer modellerin eğitimi ile bir ilişkileri bulunmamaktadır. Örneğin, dil modeli eğitirken genel bir metin verisi kullanılıp bu modelin iyileşmesi konuşma tanımayı nasıl etkileyeceği göz önünde bulundurulmuyor. Sözlük kullanan sistemlerde de, kelime okunuşları el ile belirlenip bir dildeki kelimelerin gerçek ve doğal okunuşları çoğu zaman bu sözlükte yer almıyor. En önemli bağımsızlık varsayımı ise saklı markov modellerin (hidden markov model - HMM) eğitiminde yapılıp çıktı sembollerinin birbirinden ilişkisiz olduğu varsayılıyor. Derin sinir ağlarını akustik model olarak kullanan yöntemlerde de çerçeve (frame) seviyesindeki çapraz entropi (cross entropy) fonksiyonu optimize edilmeye çalışılıyor [6, 7]. Ancak, çerçeve seviyesindeki hatanın kelime hata oranı (word error rate - WER) ile doğrudan bir ilişkisi bulunmuyor [8].

Farklı ve birbirinden bağımsız bu modelleri ilişkilendirip tek parça haline getirmek için farklı çalışmalar yapılmıştır. Tekrarlanan sinir ağları (recurrent neural network - RNN) tabanlı dil modellerinde markov zinciri varsayımı bulunmayıp tanıma çıktısındaki *n-best* listesini tekrar skorlamak için kullanılmıştır [9]. Akustik model tarafında da dizi eğitimi (sequence training) tekniği dizi seviyesindeki objektif fonksiyonu kullanarak akustik model ile kelime hata oranı arasındaki farkı kapatmak için kullanılmıştır [10, 11]. Uçtan uca bağlantıcı zamansal sınıflandırıcılar da (end-to-end connectionist temporal classification - CTC) kelime okunuşları ile akustik

modeli aynı zamanda eğitip kelimelerin gerçek okunuşlarını ses verisinden öğrenmek için kullanılmıştır [12].

Bütün bu çalışmalara rağmen bu modeller hala tek başlarına eğitilip kullandıkları eğitim verisindeki dağılımlar hakkında ayrı ayrı varsayımlarda bulunuyorlar. HMM ve CTC tabanlı sistemler markov varsayımlarını kullanan *n-gram*'ları kullanıyorlar ve diğer modellerden tamamen bağımsız olarak eğitilmekteler [1, 5, 13, 14]. Ayrıca, HMM ve CTC tabanlı sistemler çıktıdaki semboller arasında bir ilişki olmadığını varsayıyorlar ancak konuşmadaki fonemler arasında ilişki bulunup birbirlerine bağımlılar [1, 5, 12]. Dolayısıyla, HMM ve CTC tabanlı sistemler genelde büyük bir dil modeli kullanarak deşifreyi gerçekleştirip kısıtlı hafızası olan cihazlar üzerinde kullanmaya uygun değillerdir.

Bu bağımsızlık ve varsayımları ortadan kaldırıp ses sinyalini direkt Türkçe metine çeviren bir uçtan uca sinir ağının yapısı bu tezde sunulmuştur. Ayrıca, konuşma tanımada kullanılan ve birbirinden ayrı eğitilen modeller tek bir model olarak eğitilip sistemin yapısı oldukça basitleştirilmiştir. Dolayısıyla, veriler arasında herhangi bir markov varsayımı bulunmayıp sistemin doğruluk oranını etkileyen tüm bileşenler birlikte (jointly) eğitiliyor.

1.2. Tez Çalışmasının Temel Katkıları

Bu tezin ana katkıları aşağıdaki gibidir:

Türkçe için uçtan uca bir konuşma tanıma modeli: Uçtan uca eğitilen ve okunuş modeli, dil modeli ve akustik modelini tek bir model olarak öğrenen sinir ağı tabanlı bir sistem. Sondan eklemeli dillerde tanıma sonucunu iyileştiren kök-ek tabanlı bir yöntem de çıktıların üretilmesi için kullanılmıştır. Kısıtlı ses derlemi ile yeterince iyi bir model elde etmek için de transfer öğrenme yöntemi uygulanmıştır. Ayrıca, konuşma tanıma çıktısını formatlamak ve noktalama işaretlerini restore etmek için de tekrarlanan bir sinir ağı yapısı önerilmiştir.

Türkçe için eğitim derlemi: Gelecekteki çalışmalara katkı sağlayacak veri derlemleri hazırlanmıştır. Akustik model eğitimi için, yazıya dökülmüş Türkçe konuşmalardan oluşan yaklaşık 20 saatlik veri içeren konuşma derlemi hazırlanmıştır. Ayrıca, dil modeli ve okunuş model eğitimleri için kullanılabilecek Türkçe metin derlemi ve okunuş modeli bu tez kapsamında elde edilmiştir.

2. KONUŞMA TANIMA

Bu bölümde, konuşma tanıma sistemlerinin genel yapı ve işleyişi özetlenip kullanılan yöntemler ve modeller ile ilgili bilgi verilecektir. Bu sistemlerin mimarisi, modülleri ve kullanılan algoritmalar tartışılıp, bu teknolojinin uygulama alanları ve doğruluk oranı ölçümü için kullanılan metrikler özetlenecektir. Ayrıca bu sistemlerde, kullanılan HMM/GMM, HMM/DNN ve CTC tabanlı yöntemlerin alt yapısı da incelenecektir.

Türkçe, konuşma tanıma teknolojisi açısından kısıtlı kaynak (under-resourced) diller arasında bulunmaktadır [15]. Bu dil üzerinde yapılan akademik çalışmalar ve ticari uygulamaların sayısı İngilizce'ye göre daha az olup hem ticari anlamda hem de akademik olarak üzerinde çalışılması gerekmektedir. Ayrıca, Türkçe diline özgü bazı özellikler de bu dili konuşma tanıma teknolojisi açısından İngilizce'ye göre daha zor kılmıştır. Sondan eklemeli dillerde kelime sayısının yüksek olması sistemin çok daha fazla kelime arasından doğru kelimeyi tahmin etmesi gerektiğini sağlıyor [16, 17]. Diğer bir taraftan da, Türkçe için hazırlanan ve yayınlanan yazılandırılmış konuşma derlemi (transcribed speech corpus) ve yazı derlemleri (text corpus) oldukça kısıtlıdır. Özellikle son zamanlarda yüksek performans sergileyen derin sinir ağlarının bu alanda kullanılabilmesi için büyük ölçüde (binlerce saat seviyesinde) eğitim verisine ihtiyaç duyulmaktadır.

Türkçe için konuşma derlemi hazırlama çalışmaları ilk olarak [3] çalışması ile başladı. Bu çalışmada, gürültüsüz ve 140 farklı kişinin sesinden oluşan 8 saatlik bir konuşma derlemi toplanmıştır. Diğer bir çalışmada da [2] Türkçe radyo haberleri belli zaman aralıklarında kaydedilip daha sonra insanlar tarafından dinlenerek [18] yaklaşık 120 saatlik eğitim verisi elde edilmiştir. Bu derlemden eğitilen gauss dağılımı tabanlı modelin (HMM/GMM) deney seti üzerindeki kelime hata oranı %21.3 olarak raporlanmıştır.

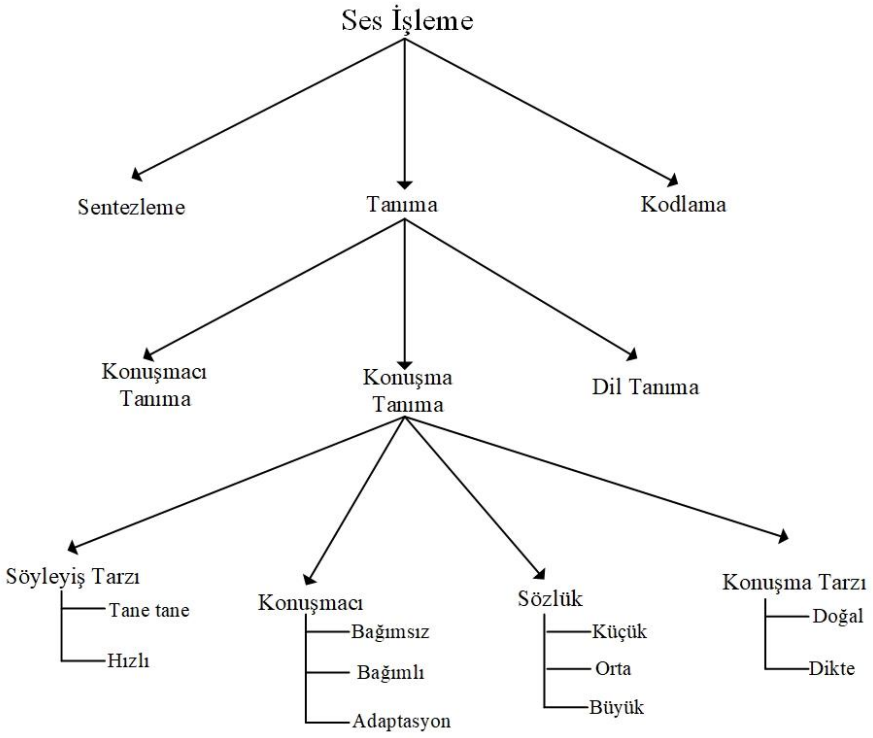
2.1. Konuşma Tanıma Sistemlerinin Çeşitleri

Konuşma tanıma sistemleri farklı özellikleri ve kullanım alanları açısından çeşitli kategorilere ayrılabilir. Bu kısımda konuşmacıya bağımlılık, sözlük boyutu, konuşma tarzı ve konuşma türü açısından bu sistemler incelenecektir. Şekil 2.1'de genel olarak ses işlemedeki süreç ve konuşma tanımadaki sınıflandırma verilmiştir.

Ayrıca, konuşma tanıma sistemlerinin kullanım alanları ile ilgili de Çizelge 2.1'de özet bilgiler verilmiştir.

2.1.1. Konuşmacıya bağımlılık

Konuşma tanıma sistemleri konuşmacıya bağımlı veya bağımsız şeklinde iki ayrı kategoriye ayrılabilir [19]. Bu alanda yapılan ilk çalışmalar ve deneyler konuşmacıya bağımlı sistemler üzerine yapılıp sistem sadece kısıtlı sayıda kişilerin sesini tanıyabiliyordu. Bu sistemlerin akustik modeli HMM/GMM – HMM/DNN tabanlı olup model eğitimi tek veya kısıtlı sayıda kişinin sesinden yapıldığı için sadece o insanların sesini tanıyabiliyor.



Şekil 2.1 Ses işlemdeki sınıflar.

Konuşmacıya bağımlı sistemler genel bir konuşma tanıma çözümü olmayıp çeşitli kişilerin sesini tanıyabilecek bir uygulama için uygun değildir. Ancak sistem sabit bir kişi tarafından kullanılacaksa ve sadece bu kişinin sesi üzerine yüksek bir doğruluk oranı bekleniyorsa bu tür durumlarda konuşmacıya bağımlı bir sistem eğitilebilir.

Konuşmacı adaptasyonu (speaker adaptation) yöntemi ile genel bir model tek bir kişinin sesine yönelik adapte edilip konuşmacıya bağımlı bir sistem elde edilebilir [20, 21, 22, 23, 24]. Örneğin, dikte programları kullanıcıya yönelik adapte edilerek tek bir konuşmacının sesini yüksek doğruluk oranında yazıya çevirebiliyor. Bu yöntem genelde cep telefonlarında veya akıllı ev asistanlarında kullanılan modeller üzerinde uygulanarak, kullanıcı veya ortama yönelik modeller eğitilip sistemin doğruluk oranı artırılmaktadır.

2.1.2. Sözlük Boyutu

HMM tabanlı konuşma tanıma sistemlerinin tanıması gereken kelimeler okunuş sözlüğü içerisinde tutuluyor. Bir kelimenin tanınması için o kelimenin hem sözlükte bulunup hem de okunuşunun model tarafından deşifre (deocde) edilebilmesi gerekmektedir. Dolayısıyla, sistemin kullanım alanına göre sözlükte bulunması gereken sözcük sayısı değişiklik gösterip bu sayı birkaç kelimeden yüz binlerce kelimeye kadar değişebilir.

Komut kontrol sistemleri (robotlar ve ev aletleri gibi sistemler) veya gömülü sistemlerde kullanılan ve sadece birkaç kelimeyi tanıması beklenen sistemler küçük sözlüklü (small vocabulary) konuşma tanıma sistemi olarak biliniyor. Bu sistemlerin tanıma oranı oldukça yüksek olup (genelde %90'ın üstündedir) sadece sözlüğünde bulunan birkaç kelimenin tanınması bekleniyor.

Bağlamı ve kullanım alanı belli olan bir konuşma tanıma sisteminin sözlüğünde genelde yüz ile iki bin arası kelime bulunup ora boyut sözlüklü (medium vocabulary) sistemler olarak adlandırılıyor. Örneğin, tıbbı ve hukuki alanlarda kullanılan bir konuşma tanıma sistemi orta boyutlu bir sözlük ile iyi bir tanıma oranı sergileyebiliyor. Bağlamı belli olmayan ve genel bir konuşma tanıma çözümü olarak kullanılacak bir sisteminin sözlüğünde yüzbinlerce kelimenin bulunması gerekmektedir. Bu sistemler büyük sözlüklü konuşma (large vocabulary speech recognition - LVSR) tanıma sistemleri olarak adlandırılıp diğer türlere göre doğruluk oranı daha düşüktür. Örneğin, dikte programları ve haber arşivlerini yazıya döken sistemlerin sözlüğü milyonlarca kelimenin okunuşunu içerebiliyor.

Sözlükte bulunan kelime sayısı ile sistemin hata oranı arasında bir denge bulunması gerekmektedir. Sözlük dışı kelimeler (out-of-vocabulary - OOV) sistem tarafından tanınmayıp kelime hata oranının yükselmesine neden oluyor. Diğer taraftan da,

sistemin kullanım alanı dışındaki kelimelerin sözlükte bulunması modelin karar vermesini zorlaştırıp benzer kelimeler arasında hata yapma olasılığını yükseltiyor. Dolayısıyla, bu sistemlerin sözlüğü hem bağlama uygun seçilip hem de insan tarafından kontrol edilerek optimize edilmesi gerekiyor. Ayrıca, bir kelimenin birden fazla okunuşu varsa tüm bu okunuşların da sözlükte bulunup kontrol edilmesi gerekiyor [1, 8, 25].

2.1.3. Konuşma Tarzı

Kullanıcı sisteme spontane (spontaneous speech recognition) veya dikte yapma tarzında konuşabilir. Doğal konuşma tarzını tanıyan bir sistem, konuşmacının farklı hızlardaki konuşmasını yüksek doğrulukta yazıya çevirmesi gerekiyor. Konuşmacı bir sistem ile iletişimde olduğunu göz önünde bulundurmadan ve herhangi bir kısıtlama olmadan spontane bir şekilde bu sistemlerle konuşabilmesi gerekiyor. Yapılan konuşma tane tane veya önceden hazırlanan bir metin üzerinden yapılıyorsa sistemin doğruluk oranı daha yüksek olmaktadır. [26, 27].

2.2. Anahtar Kelime Yakalama Sistemleri

Anahtar kelime yakalama sisteminin amacı, bir ses akışı üzerinde daha önceden belirlenen kelimelerin yakalamasıdır (keyword spotting). Bu sistemler otomatik kelime uyarısı yapan uygulamalarda, çağrı merkezi konuşmalarında ve akıllı telefonlarda yaygın olarak kullanılmaktadır. Sistem sürekli olarak bir ses akışını dinleyip kullanıcı tarafından belirlenen kelime veya kelimeler geçtiği an tetikleniyor. Örneğin, akıllı Android cihazlarında “**Okay Google**” denildiği zaman cihaz dinleme durumuna geçip konuşmacının komutlarını yapmaktadır.

Kelime yakalama sistemleri çevrim-dışı (offline) veya çevrim-içi (online) olarak iki farklı durumda çalışıyorlar. Çevrim-dışı bir sistem, ses kaydını alıp konuşma tanıma yaptıktan sonra indekslenen kelime listesi üzerinde arama yapıyor. Bu tür sistemde konuşma tanımanın doğruluk oranı ve üretilen latislerin (lattice) kalitesi kelime yakalamanın başarısını etkileyip büyük veri kümeleri kullanarak HMM/GMM [28, 29, 30] veya HMM/DNN [31, 32] model eğitimleri yapılıyor. Çevrim-içi çalışan sistemler ise, ses akışı canlı olarak dinlenip sadece anahtar kelimeler aranıyor. Bu tarz sistemlerde bütün ses kaydı üzerinde bir konuşma tanıma yapılmayıp sadece belirlenen kelimeler aranıyor.

2.3. Konuşmacı Tanıma ve Doğrulama

Konuşmacı tanıma sisteminin görevi verilen bir sesin hangi kişiye ait olduğunu bulmaktır. Farklı kişilerin ses örnekleri model olarak sisteme tanıtılıp (enrollment) daha sonra gelen bir ses örneği bu veri tabanı üzerinde aranıyor. Bu sistemler, metinden bağımsız veya metine bağımlı olarak HMM/GMM [33, 34] veya DNN [35, 36, 37, 38] modelleri ile eğitilebiliyor. Metine bağımlı modelde sadece sabit bir metinin hem kayıt aşamasında hem de tanıma zamanında okunması gerekiyor. Metinden bağımsız sistemlerde ise okunan metin önem taşımayıp serbest bir konuşmada kullanıcının kimliği tespit edilmektedir.

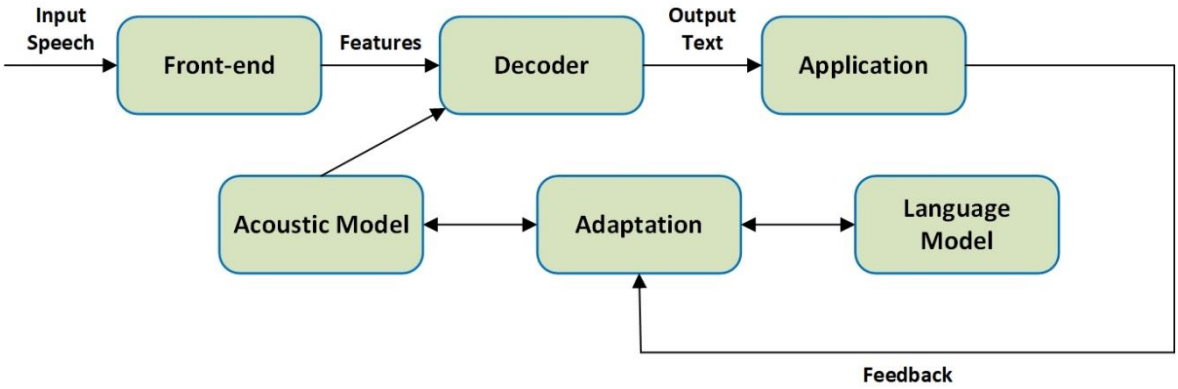
Konuşmacı doğrulama sisteminin amacı bir ses örneğinin gerçekten iddia edilen kişiye ait olup olmasını tespit etmektir. Sesli şifre, adli bilişim ve güvenlik sistemlerinde konuşmacı doğrulama sistemleri yaygın olarak kullanılıp bu sistemin doğruluk oranı oldukça önem taşımaktadır. Doğrulanması istenilen kişilerden ses örnekleri alınarak sistem tanıtılıp daha sonra gelen bir ses bu model ile karşılaştırılarak benzerlik oranı hesaplanıyor. Elde edilen benzerlik oranı eşik değerinin üstündeyse, ses sistem tarafından doğrulanıp aksı durumda reddediliyor. Modelleme için HMM/GMM [39, 40] ve HMM/DNN [35, 41, 42, 43, 44, 45] tabanlı modeller hem metinden bağımsız hem de metine bağımlı konuşmacı doğrulama sistemlerinde kullanılmaktadır.

Çizelge 2.1 Konuşma tanıma teknolojisinin kullanıldığı alanlar ve uygulamalar.

Kullanım Alanı	Uygulama	Girdi	Çıktı
İletişim sektörü/Telefon	Telefon rehberinden sesli sorgulama	Ses sinyali	Konuşulan kelimeler
Eğitim Sektörü	Öğrencilere yabancı dil öğretmede kelimelerin doğru okunuşunu öğretmek. Özürlü insanların klavye kullanmadan bilgisayar ile iletişim kurmaları	Ses sinyali	Konuşulan kelimeler
Beyaz eşya	Fırın, çamaşır makinesi ve buzdolabı gibi ev eşyalarında konuşma tanıma	Ses sinyali	Konuşulan kelimeler
Savunma sanayi	Savunma sanayide kullanılan araçların kontrol edilmesi	Ses sinyali	Konuşulan kelimeler
Yapay zeka	Robotların kontrolü için	Ses sinyali	Konuşulan kelimeler
Sağlık sektörü	Doktorların muayene raporlarını yazmak	Ses sinyali	Konuşulan kelimeler
Çağrı merkezleri	Telefon görüşmelerinin otomatik yazıya dökülmesi ve kalite kontrol	Ses sinyali	Konuşulan kelimeler

2.4. Sistemin Mimarisi

Konuşma tanıma sisteminin görevi ses sinyalini birtakım algoritma ve modelleri kullanarak işleyip metine dönüştürmektir. Şekil 2.2'de bu sistemlerin mimarisi ve kullanılan bileşenlerin genel şeması çizilmiştir. Klasik bir konuşma tanıma sisteminin mimarisinde üç önemli bileşen bulunmaktadır; sistemin giriş noktası olan ilk aşamada, örüntü tanıma (pattern recognition) açısından önemli olan öznitelik vektörleri (feature vectors) ön-uç bileşeni tarafından hesaplanıyor [46, 47]. Daha sonra, deşifre (decoder) modülü akustik model, dil modeli ve sözlüğü kullanarak elde edilen öznitelik vektörlerini fonem dizilerine dönüştürmektedir.



Şekil 2.2 Konuşma tanıma sisteminin mimarisi.

Bu bileşenlerin her birisi farklı model ve eğitim verisi ile eğitilip bütün bunların birlikte çalışması sonucunda girdideki ses sinyali harf veya kelime dizisine dökülüyor. Bu mimarinin gerçekleştirilmesi için, elektrik mühendisliği, bilgisayar bilimleri, bilgisayar mühendisliği ve dil biçimsel gibi farklı uzmanlık alanlarının buluşması gerekiyor.

2.4.1. Ön Uç

Konuşma tanıma sisteminin giriş noktası ön uçtur. Bu modül, konuşma sinyalindeki gerekli öznitelik vektörlerini çıkartıp (feature extraction) sinyale ait daha özet ve yararlı bilgi sunmaktadır. Elde edilen öznitelik vektörleri fonemleri tanımak için yeterli olup konuşma sinyalindeki gereksiz bilgiler bu vektörler içerisinde bulunmamaktadır. Konuşma tespiti (speech detection), gürültü azaltma (noise reduction) gibi aşamalar bu bileşenin görev listesinde ölüp sistemin doğruluk oranını büyük ölçüde etkilemektedir.

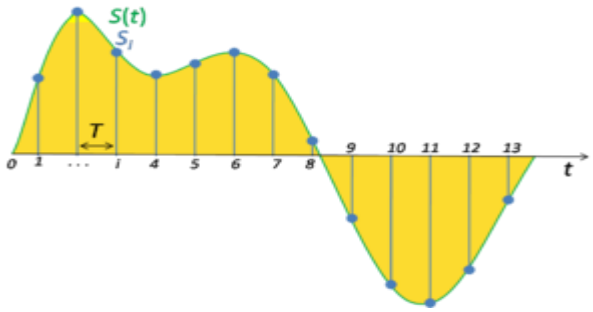
2.4.1.1. Ön İşleme:

Ön-uç modülü, konuşma sinyali üzerinde birtakım ön işlemeler (speech pre-processing) uygulayıp model eğitimi ve deşifre için gerekli olan sinyal işleme adımlarını gerçekleştiriyor. Bu görevlerden birisi; konuşma sinyalinde bulunan ve tanıma için gerekli olmayan sesler ve gürültülerin konuşma sinyalinden temizlenip çıkartmaktır. Gereksiz seslerin girdiden temizlenmesi daha az veri üzerinde işlem yapılacağı anlamına gelip dolayısıyla sisteminin verimliliğini de artırmaktadır. Ses aktivasyon tespiti (voice activity detection - VAD) algoritması [48] ön uç aşamasında devreye girip sessizlik kısımları otomatik olarak etiketleyerek konuşma sinyalinden

çıkarabilmektedir. Bu sayede, deşifre modülüne sadece konuşma içeren ses parçaları gönderebilip sisteminin dönüş süresi (response time) kısaltılmaktadır [27].

2.4.1.2. Öznitelik Çıkartma:

Konuşma sinyalinin, bilgisayarlar ve örüntü tanıma algoritmaları için kullanılabilir olması için dijital örneklere dönüştürülmesi gerekmektedir. Analog ortamdaki sinyal, mikrofon vasıtası ve ses kartı ile dijital örneklere aktarılmaktadır. Bu aktarım sürecinde, analog bir sinyalinin belli zaman aralıklarında belli sayıda örnekleri alınıp dijital sayılar şeklinde kaydediliyor. Dijitalleştirme sonucunda, analog ortamdaki sürekli (continuous) olan bir sinyal dijital ortamda kesikli (discrete) sinyal örneklerine dönüştürülüyor. Bu örnekleme süreci iki farklı parametre ile ifade edilip birincisi örnekleme sıklığı (sampling frequency) ikincisi de örneklerin tutulmasında kullanılan bit sayısıdır (Şekil 2.3). Bu şekildeki *s(t)* fonksiyonu örneklenmesi istenilen sürekli bir sinyali gösterip ve örnekleme işlemi bu fonksiyonun *T* zaman aralığındaki değerleridir. Ayrıca örnekleme sıklığı f_s ile gösterilip, bir saniye içerisinde *s(t)* fonksiyonundan alınan örnek sayısını ifade etmektedir.



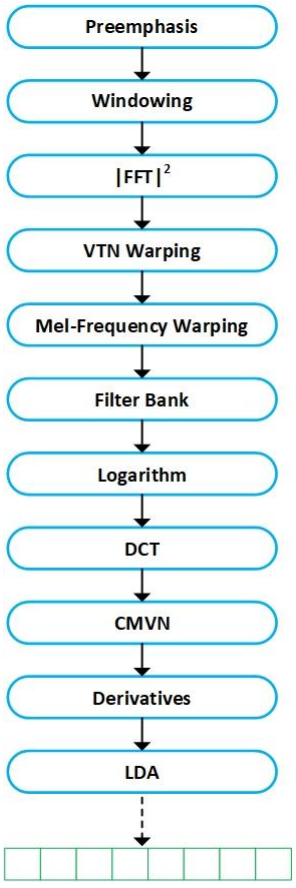
Şekil 2.3 Analog sinyalin dijitalleşmesi.

Telefon hatları ve kablosuz mikrofonlar ile alınan kayıtların örnekleme sıklığı 8 Khz olup konuşmaların anlaşılabilir olması için yeterlidir. Ancak, daha kaliteli ses kayıtları için 11025 Hz, 16000 Hz, 22050 Hz ve 44100 Hz gibi daha yüksek sayılarda örnekleme sıklıkları da yaygındır.

Örnekleme süreci zaman boyutunda yapıldığı için konuşmadaki harflerin veya kelimelerin tanınması için uygun bir gösterim şekli değildir. Örneğin, 16 Khz'de kayıt edilen bir ses dosyasının her bir saniyesinde 16 bin örnek bulunuyor ancak,

konuşma sinyalindeki fonem ve kelime örüntülerinin tanınması için bu örnekler gerekli bilgiyi içermemekteler [49]. Zaman ve genlik (amplitude) boyutundaki sinyal bilgileri fonem ve kelimelerin modellenmesi için değerli bir bilgi taşımayıp frekans boyutundaki bilgiler önem taşımaktadır. Bu yüzden de, öznitelik çıkarmanın en önemli amacı frekans boyutundaki bilgileri kullanarak girdi vektörlerini oluşturmaktır. Mel Frequency Cepstral Coefficient (MFCC) [46] ve Perceptual Linear Prediction (PLP) [47] gibi yöntemler öznitelik çıkarsama için kullanılabilir.

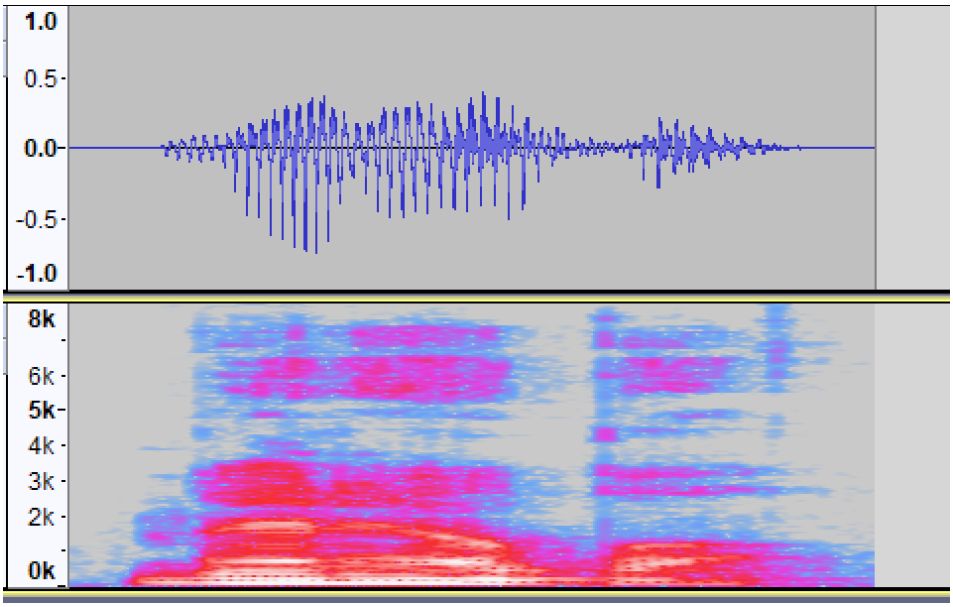
MFCC vektörleri, konuşma tanıma alanında öznitelik çıkama için kullanılan yaygın yöntemlerden birisidir. Bu öznitelik vektörleri, sinyaldeki örnek sayısını oldukça düşürüp fonemlerin tanınması için gerekli olan bilgileri kendisinde barındırıyor. MFCC hesaplamasında izlenmesi gereken adımlar sırasıyla Şekil 2.4'te verilmiştir.



Şekil 2.4 MFCC öznitelik vektörlerinin hesaplanması için izlenen adımlar.

Fonemleri ve harfleri birbirinden ayırt eden özellik bunlara ait sesteki enerji miktarı ve enerjilerin biriktiği frekanstır. Dolayısıyla, ses sinyalini Fast Fourier Transform (FFT) kullanarak zaman boyutundan frekans boyutuna taşıyıp gerekli ve yararlı bilgiler elde edilmektedir.

Şekil 2.5'te "*merhaba"* kelimesi için waveform ve Spectrogram çizimleri gösterilmiştir. Bu şekilde de görüldüğü üzere, waveform sadece bir sinyalin zaman boyutundaki değişimlerini gösterip frekans ile ilgili herhangi bir bilgi içermemektedir. Bu yüzden de, waveform gösterimi ses sinyalinden fonem ve harflerin çıkartılması için yeterince bilgi içermiyor. Spectrogram çiziminde daha koyu renk ile gösterilen kısımlarda, enerji seviyesi diğer kısımlara göre daha yüksek olup enerjinin hangi frekanslarda biriktiği gözükmektedir.

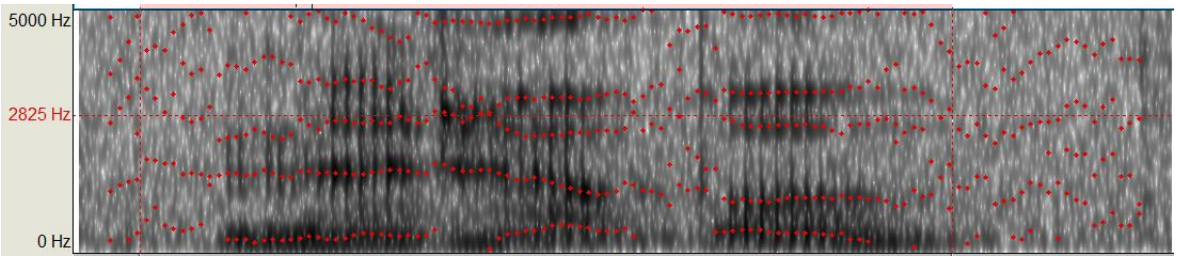


Şekil 2.5 "merhaba" kelimesinin Waveform ve Spectrogram gösterimi.

2.4.1.3. Formant

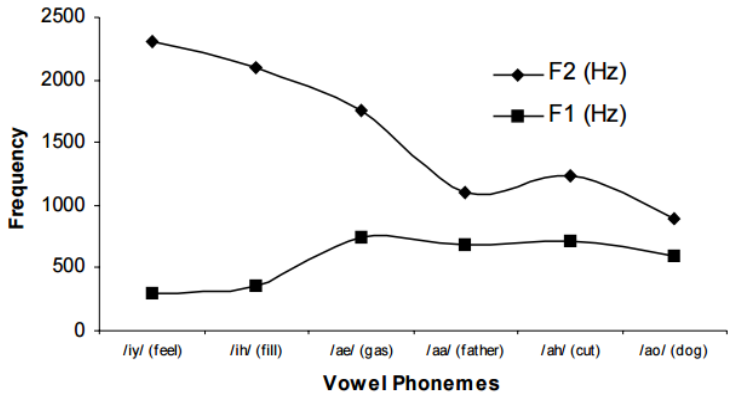
Ses sinyalinde bulunan frekans spektrumundaki maksimum değerleri Formant olarak adlandırılmaktadır. Konuşma analizinde, Formant kavramı gırtlaktaki tınlaşım (resonance) olarak bilinip ses sinyalini oluşturan belirgin frekans bileşenleridir. Ayrıca, harflerin tanınması için en önemli özniteliklerden birisi de Formant bilgisidir.

En düşük frekansta bulunan Formant F1 ile, ikinci seviyede bulunan Formant F2 ve üçüncü seviyede bulunan Formant ise F3 ile ifade edilir. Formant değerleri Spectrogram'daki enerjinin biriktiği yerlerdeki çizgiler halinde ortaya çıkıp Şekil 2.6'da daha koyu renkler ile görülebilir. Genelende, F1 ve F2 değerleri sesli harflerin tanınması için yeterli olup sessiz harflerin tanınması için ise F3 değerleri daha önemli bilgiler taşıyor.



Şekil 2.6 "merhaba"a kelimesinin Formant bilgileri.

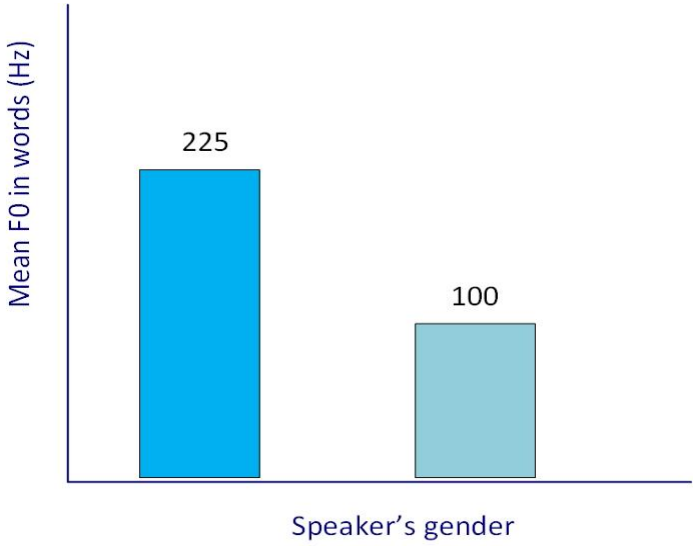
İngilizce'deki bazı sesli harfler için F1 ve F2 değerlerinin ortalaması Şekil 2.7'da çizilmiştir.



Şekil 2.7 Bazı İngilizce fonemlerde F1 ve F2 değerleri.

Konuşmacı tanıma (speaker identification) ve doğrulama (speaker verification) alanlarında Formant değerleri önemli etken olarak göz önünde bulunduruluyor. İnsanların gırtlak yapısındaki farklılıklar, farklı Formant değerlerinin üretilmesine yol açıp seslerin birbirinden ayırt edilebilmesi için önemli etkenlerden birisidir. Dolayısıyla, ses biyometrisi (voice biometrics) gibi alanlarda Formant

değerlerinin doğru hesaplanması ve kullanılması oldukça önem taşımaktadır. Örneğin, Şekil 2.8'de iki heceli kelimelerdeki F0 değerinin ortalaması kadın ve erkek sesi için çizilmiştir. Görüldüğü üzere kadın sesindeki F0 değerinin ortalaması 225'iken erkek sesindeki F0 değerinin ortalaması 100 civarındadır.



Şekil 2.8 Kadın ve Erkek sesindeki F0 değerinin maksimumu.

2.4.2. Modeller

Konuşma tanıma sistemlerinde akustik model (Acoustic Model) ve dil modeli (Language Model) olarak bilinen iki model bulunmaktadır. Akustik modelin görevi kelimelerdeki akustik özellikleri modelleyip her bir fonemi matematiksel bir model ile göstermektir. Akustik model eğitimi, konuşma örnekleri ve bunlara ait birebir metini kullanarak yapılıyor. Dil modeli ise bir dildeki kelimelerin dizilişi ve cümlelerin biçimsel yapısını modelleyip konuşma tanıma sisteminin doğruluk oranını yüksek derecede etkilemektedir [1]. Bu model, büyük miktarda metin derlemlerinden eğitilip istatistiklerin gerçekçi olması açısından metinlerin çeşitli konulardan toplanması gerekiyor.

Bu kısımda, HMM tabanlı konuşma tanıma sistemlerinin yapısı matematiksel olarak ele alınacaktır. Konuşma tanıma sisteminin asıl amacı **x** sinyalini **w** kelime dizisine çevirmektir. Bayes kuralını uygulayarak bu olasılığı akustik model $\mathbf{P_{AM}(x \mid w)}$ ve dil modeli $\mathbf{P_{LM}(W)}$ olarak iki farklı bileşene ayırabiliriz:

$$P(w|x) = \frac{P(x|w)P(w)}{P(x)} \propto P(x|w)P(w) = P_{AM}(x|w)P_{LM}(w) \quad (2.1)$$

2.4.2.1. Akustik Model

Akustik model $\mathbf{P_{AM}(x|w)}$ DNN ve HMM olarak iki modelden oluşmaktadır. İlk olarak x_t sinyal çerçevesindeki q_t fonemine ait sonsal olasılık hesaplanıyor:

$$\mathrm{P}(\mathrm{q_t}|\mathrm{x_t}) = \mathrm{DNN_{AM}}(\mathrm{x_t}) \quad (2.2)$$

Veya tekrarlanan sinir ağı akustik model olarak kullanılırsa [50]:

$$\mathrm{P}(\mathrm{q}|\mathrm{x}) = \mathrm{RNN_{AM}}(\mathrm{x}) \quad (2.3)$$

t zamanı için bir fonemin benzerlik (likelihood) olasılığını hesaplamak için sonsal olasılığı basit bir şekilde önsal olasılığa bölerek elde edebiliriz:

$$\mathrm{P}(\mathrm{x_t}|\mathrm{q_t}) = \frac{\mathrm{P}(\mathrm{q_t}|\mathrm{x_t})\mathrm{P}(\mathrm{x_t})}{\mathrm{P}(\mathrm{q_t})} \quad (2.4)$$

$P(x_t)$ kelime dizisinden bağımsız olduğu için göz ardı edilebilir [51].

Fonem sırası HMM'deki Triphone yapısına birebir eşleştirilebilir:

$$P(q_{t+1} = u|q_t = v) = \beta(u, v) \quad (2.5)$$

u ve *v* HMM'deki durumları (State) gösterip β ise geçiş matrisidir [52].

Genelde, fonemlerin hizalanması (phoneme alignment) q_t normal gauss dağılımını kullanarak (HMM/GMM) elde ediliyor [7]. HMM/GMM sistemi Expectation Maximization (EM) algoritması ile eğitilip daha sonra bu modeli kullanarak hizalama yapılıp DNN sistemi eğitiliyor [51, 53, 54, 55]. Bu yöntemde DNN modelin başarısı

GMM sistemin ürettiği hizalamanın kalitesi ile orantılı olup bazı çalışmalarda bu hizalamalar el ile düzeltilmiştir [55].

2.4.2.1.1. Markov Varsayımı

HMM'de, durumlar arasındaki geçişlerde markov varsayımları yapılıp bir durumdan başka duruma geçişte sadece bir önceki durum göz önünde bulunduruluyor:

$$p(q_t|q_{<t}) = p(q_t|q_{t-1}) \quad (2.6)$$

Bu kısıtlama da uzun bağımlılığı olan dizilerin modellenmesini imkansız kılıyor. Örneğin, fonem sayısı fazla olan kelimelerin modellenmesi Markov varsayımına uymuyor.

Ayrıca, HMM'deki durumların yayım olasılığı da (emission probability) birbirinden bağımsızdır [56]. Çerçeveler arası (frame) yayım olasılığı q_t durumundan bağımsızdır:

$$p(x_t|x_{<t}, q_{<t}) = p(x_t|q_t) \quad (2.7)$$

DNN tabanlı akustik modelde ise fonemlerin sonsal olasılığı bir pencere için bağımsızdır:

$$p(q_t|x, q_{<t}) = p(q_t|x_t) \quad (2.8)$$

Bu bağımsızlık kelimede bulunan fonemlerin birbirinden bağımsız olması anlamına geliyor. Ancak, bu varsayımın yanlış olup fonemler arasında istatistiksel bağımlılık bulunuyor.

2.4.2.1.2. Çapraz Entropi Hatası

DNN ve RNN akustik modeller genelde çapraz entropi veya pencere seviyesindeki hata fonksiyonlarını minimize ederek eğitiliyorlar. Ancak, pencere seviyesindeki hata oranının düşmesi her zaman kelime hata oranının düşeceği anlamına

gelmemektedir [57]. Dizi eğitimi [11] (sequence training) bu sorunun ortadan kaldırılması için önerildi. Fakat, bu yöntem sadece akustik model tarafını iyileştirip okunuş ve dil modeli tarafında iyileşme yapılmıyor.

2.4.2.2. Sözlük

Sözlük içerisinde sistemin tanıması gereken sözcükler yer alıp sistemin kullanım alanına göre sözlükteki kelime sayısı değişebiliyor. Büyük sözlüklü otomatik konuşma tanıma sistemlerinde (large vocabulary automatic speech recognition system - LVASR) yüzbinlerce kelimenin farklı okunuşları yer almaktadır.

Sözlük hazırlama aşaması, HMM tabanlı sistemlerinin tasarlanması sürecindeki önemli adımlardan birisidir. Sözlük dışı kelimelerin azaltılması ve gereksiz kelimelerin sözlükte bulundurulmaması arasında bir denge kurulması gerekiyor. Okunuş sözlüğündeki tüm kelimelerin okunuşu (pronunciation) fonetik semboller ile belirlenip her bir HMM bu sembollerden birisini ifade ederek modellemektedir. Türkçe, yazıldığı gibi okunan bir dil olduğu için sözlük hazırlama süreci daha kolay olup kelimedeki harfler fonetik sembol olarak da kullanılabilir.

Şuana kadar anlatılan akustik model **x** sinyaline ait **q** fonem dizisini üretebiliyor. Ancak konuşma tanıma sisteminde fonem dizisi değil kelimeleri çıktı olarak elde etmek istiyoruz. Okunuş sözlüğü sistemi doğru okunuşları üretmesi için zorlayıp ayrıca fonem çıktısını olası ve doğru kelimelere çevirerek beklenen kelime çıktısını da üretiyor.

Aynı kelime farklı şekillerde okunabildiği için sözlükte bulunan her kelimenin çeşitli okunuş şekilleri fonetik olarak belirtilmelidir. Ayrıca, yabancı kelimeler ve kısaltmalar gibi kelimelerin de yazılışı ile okunuşları farklı olabileceği için bütün bu okunuşlar sözlükte bulunmalıdır:

Word	Pronunciation	
Merhaba	M E R H A B A	M E R A B A
Nasılsınız	N A S I L S I N I Z	N A S I S I N I Z
TTNET	T E T E N E T	T İ T İ N E T
Mobile	M O B İ L	M O B A Y L
?	S O R U	S O R İ

Genel bağlam için hazırlanan sözlükler genelde büyük metin derleminden elde edilen tekil kelime listesinden oluşuyor. Kelimeler dil bilim uzmanları tarafından incelenip farklı okunuş ve yazılışları ile sözlüğe dahil ediliyor. Bu tezde, dil modeli kullanılan deneylerde metin derlemindeki tekil kelimeler sözlük olarak kullanılmıştır. Dil modeli ve sözlük oluşturmak için daha çok Türkiye Büyük Millet Meclisi'ndeki (TBMM) tutanak metinleri kullanılmıştır. Bu tutanaklar farklı insanlar tarafından kontrol edildikleri için imla yanlışlığı olan kelime sayısı diğer kaynaklara (haber siteleri, e-kitaplar) daha az olup model eğitimi için daha uygundur.

2.4.2.3. Dil Modeli

Dil modeli $P_{LM}(w)$ dildeki kelimelerin dizilişini akustik özelliklerinden tamamen bağımsız olarak modelleyip ve genelde *n-gram*'ları modelleme yöntemi olarak kullanmaktadır [4]. Bu Modeller de Markov varsayımında bulunup genelde 2 ile 4 arası geçmişteki kelime sırasını göz önünde bulundurarak olasılık hesaplamalarını yapıyorlar. Dolayısıyla, uzun bir cümledeki kelimelerin dizilişini modellemek *n-gram*'lar ile mümkün olmayıp sadece kısıtlı kelime geçmişi modellenebiliyor. RNN tabanlı dil modellerinde Markov varsayımı bulunmadığı için bu modeller kelimelerdeki uzun bağımlılıkları modelleyebiliyorlar. Ancak, RNN modeli deşifrede kullanmak için çok yavaş kalıp ayrıca deşifre grafına entegre edilemiyor [1]. Bu yüzden RNN'ler genelde latisleri (lattice) tekrardan skorlamak için kullanılıyor [9].

Dil Modeli, bir dildeki kelimelerin ve cümlelerin yapısı ve sırasını modelleyerek o dile ait bir istatistiksel model üretmektedir. En basit deyişle, bu model bir kelime diziden sonra hangi kelimelerin gelebileceğini modelleyip deşifre zamanında olası dizilişleri üretmektedir. Gerçekçi bir istatistik elde etmek için olasılık hesaplamaları yeterince büyük metin derlemleri kullanarak yapılmalıdır. Bu derlem genelde online gazetelerden, elektronik kitaplardan ve dijital metin içeren farklı kaynaklardan elde edilip bazı ön işlemelerden sonra eğitim verisi olarak kullanılıyor.

Cümledeki kelime sırası ve istatistikler *n-gram* modelleri ile modellenmektedir:

$$P(W) = P(w_1)P(w_2|w_1)P(w_3|w_1, w_2) \dots P(w_n|w_1, \dots, w_{n-1}) \quad (2.9)$$

Dil modelinin bağlama yakın olması bazı durumlarda sistemin kelime hata oranını düşürüyor. Örneğin, siyasi konuşmalar içeren bir konuşma arşivi üzerinde tanıma

yapılacaksa siyasi metinlerden oluşan bir derlemden dil modeli üretmek kelime hata oranını daha fazla düşürüyor. Bunun nedeni, belli bir bağlamdaki kelimelerin sırası ve yapısı doğal olarak diğer bağlamlara göre farklılık gösterebileceğinden kaynaklanmaktadır [58].

Dil modeli eğitirken metin derlemi içerisinde çok sık geçen kelime dizilerinin olasılık değeri seyrek geçen kelime dizilerine göre daha yüksek çıkıp sistemin çıktısını da bu yönde etkiliyor. Ayrıca, metin derlemi ne kadar büyük olsa da birçok *n-gram*'ın bu derlem içerisinde bulunmaması olasıdır. Bu tür sorunları çözmek için dil modeli üzerinde düzleme (smoothing) yöntemleri uygulanmaktadır [59]. Düzlemedeki amaç, olasılığı çok yüksek olan *n-gram*'ların olasılık değerini biraz düşürürken olasılığı düşük olan *n-gram*'ların ise olasılık değerini artırmaktır. Ayrıca, metin derlemi içerisinde hiç geçmeyen *n-gram*'lar için mevcut olasılıkları kullanarak (backoff weights) bir olasılık hesaplanabilir [60].

HMM tabanlı sistemlerde, dil modeli akustik modelden bağımsız olarak eğitiliyor. Ayrıca, dil modeli genelde karmaşıklığı (perplexity) düşürme yönünde eğitilip konuşma tanımadaki diğer faktörler göz önünde bulundurulmuyor. Fakat, karmaşıklığın kelime hata oranı ile her zaman orantılı bir ilişkisi bulunmamaktadır [4]. Dolayısıyla, dil modeli ve akustik modeli tek bir model olarak eğitip kelime hata oranının düşmesi yönünde bu modeli optimize etmek ideal bir durumdur.

2.4.3. Deşifre

Deşifre sırasındaki amacımız elimizdeki **x** sinyaline ait en olası **w** kelime sırasını bulmaktır:

$$\hat{\mathrm{w}} = \operatorname{argmax} p_{AM}(x|w)p_{LM}(w) \tag{2.10}$$

Ancak gerçek uygulamalarda akustik modelin tek başına kullanılması kelime hata oranını oldukça yükseltiyor. Genelde, P_{AM} oldukça zayıf bir model olup yukarıda anlatılan bağımsızlıklardan dolayı tam olarak doğru kelimeleri üretemiyor. Bu yüzden dil modeline daha fazla β ağırlığı vererek bu modelin etkisini akustik modele göre arttırıyoruz:

$$\hat{\mathrm{w}} = \operatorname{argmax} p_{AM}(x|w)p_{LM}(w)^{\beta} \tag{2.11}$$

HMM tabanlı konuşma tanıma sistemlerinin başarısı daha çok dil modeline bağımlı kalmaktadır. Dolayısıyla, oldukça güçlü bir dil modelinin eğitilmesi doğruluk oranı yüksek bir konuşma tanıma sistemi elde etmek için kaçınılmaz oluyor [1, 5].

2.4.3.1. Bağımsızlık Varsayımı

CTC algoritması, hizalama çıktısındaki bir a_t sembolü ile **x** dizisi arasında bağımsızlığı varsayıyor:

$$p(a_t|x, a_{<t}) = p(a_t|x) \tag{2.12}$$

Girdi ile çıktı arasında derin bir RNN yapısı olsa bile girdi ile çıktı arasında güçlü bir Markov varsayımı bulunuyor. Bir çerçeve üzerindeki tahmin ile komşusundaki çıktı arasında koşullu bağımsızlık var. Bu yüzden de, CTC algoritması çıktısındaki sembollerin diline öğrenemeyip deşifre aşamasında güçlü bir dil modeline ihtiyaç duymaktadır.

3. VERİ HAZIRLAMA VE TOPLAMA

Konuşma tanımadaki modellerin (akustik ve dil modeli) eğitimi için konuşma ve metin veri kümeleri kullanılmaktadır. Konuşmalar ve metin akustik model eğitimi sırasında hizalanıp konuşma sinyaline eşitlenen fonemler model tarafından öğreniliyor. Bu eğitim setinde bulunan verilerin niteliği ve niceliği modellerin doğru ve gerçekçi olmasını doğrudan etkiliyor. Çeşitli insanların, farklı yaş aralıklarında ve farklı cinsiyetten konuşmaların bulunması sisteminin konuşmacıdan ve aksandan bağımsız (speaker independent) hale gelmesini sağlıyor. Ayrıca, sistemin farklı akustik ortamlardaki konuşmaları tanıması ve ortamdan bağımsız olması için de çeşitli ortamlardan alınan ses örnekleri eğitim setinde bulunmalıdır. Örneğin, gürültülü ortamlardaki konuşmalar, telefon konuşmaları, arabada yapılan konuşmalar ve gürültüsüz konuşmalar gibi çeşitli örneklerin bu derlem içerisinde bulunması sistemi ortamdan bağımsız ve gürültüye dayanıklı (noise robust) olması için gereklidir.

Türkçe dili için yayınlanan ve akademik çalışmalarda kullanabileceğimiz kısıtlı sayıda yazıya dökülmüş konuşma ve metin derlemleri bulunmaktadır [2, 3]. [2] çalışmasında stüdyo kayıtlarından ve gürültüsüz konuşmalar içeren bir konuşma derlemi toplanmıştır. Ses kayıtları, daha önceden hazırlanan cümleleri ana dili Türkçe olan insanlara okutarak alınıp toplam sekiz saatlik kayıt elde edilmiştir. Diğer bir çalışmada [3] farklı radyo ve televizyon kanallarından kaydedilen Türkçe konuşmalar yazıya dökülüp yaklaşık 120 saatlik bir derlem oluşturulmuştur. Bu çalışmada, kaydedilen sesler önce bölütlenip (segmentation) daha sonra insanlar tarafından dinleyerek yazıya dökülmüştür.

3.1. Konuşma Derlemi Hazırlama Süreci

Tez kapsamında iki farklı yöntemle ve çeşitli kaynaklardan, yazıya dökülmüş konuşma derlemi elde edilmiştir. Bu derlem akustik model, dil modeli ve sözlük oluşturma gibi aşamalarda kullanılmaktadır. Ayrıca veri sentezleme gibi yöntemlerle de, ses veri tabanı zenginleştirilip aynı kaydın farklı versiyonları elde edilmiştir.

3.1.1. Mobil Uygulama

Birinci yöntemde, daha önceden hazırlanan Türkçe cümleler insanlara okutarak bir derlem hazırlama çalışması yapıldı. Bu yöntemde bir cep telefonu uygulaması kullanarak çeşitli cümleler farklı ortamlarda insanlara okutulup kaydedilmiştir.

Kayıtlara başlamadan, konuşmacının cinsiyeti, yaşı ve ortam bilgisi gibi sorular cevaplanıp meta veri olarak tutuluyor. Daha sonra, kullanıcı karşısına çıkan cümleleri söyleyip telefon mikrofonu ile kaydediliyor.

Kullanıcı karşısına çıkan cümleler daha önceden hazırlanan ve bin cümleden oluşan bir kümeden rastgele seçilmektedir. Tüm kayıtlar 16 Khz, 16 bit ve PCM formatında alınmıştır. Bu çalışma sonucunda toplamda 4 saatlik bir konuşma derlemi elde edilip mobil verisi olarak adlandırılmıştır.

Çizelge 3.1'de bu yöntem ile elde edilen konuşma derlemi hakkında özet bilgiler sunulmuştur. Kaydedilen konuşmaların iki saati kapalı ortamda, bir saati açık ortamlarda ve geri kalan bir saatlik kısmı ise araba içerisinde alınmıştır. Derlem içerisinde, 60 kadın ve 40 erkek olmak üzere toplamda yüz kişinin ses kaydı bulunmaktadır. Bu süreç sonucunda tüm kayıtlar dinlenip kayıtların %5'lik bir kısmı sorunlu olarak tespit edilip derlem içerisinden çıkartılmıştır.

Çizelge 3.1 Mobil uygulama ile hazırlanan konuşma derlemi.

Ortam	#Saat	#Erkek	#Kadın
Kapalı	2	30	20
Açık	1	10	10
Araba	1	20	10
Toplam	4	60	40

Kayıt işlemi için öngörülen cümleler yabancılar için hazırlanan Türkçe öğretim kitaplarından seçilmiştir. Toplamda bin cümleden oluşan bir metin derleminden rastgele seçimler yaparak kullanıcılara okutulmuştur. Her kullanıcı ortalama 50 cümleyi okuyup ses kayıtlarının ortalama uzunluğu ise 3 saniyedir. Kayıt için seçilen örnek cümleler aşağıdaki gibidir:

- *Metro istasyonuna nasıl gidebilirim*
- *İki kişilik bir oda rezervasyonu yapmak istiyorum*
- *Dün gece arkadaşlarımla dışarı çıktık ve sabahlara kadar eğlendik*

3.1.2. Web Tabanlı Uygulama

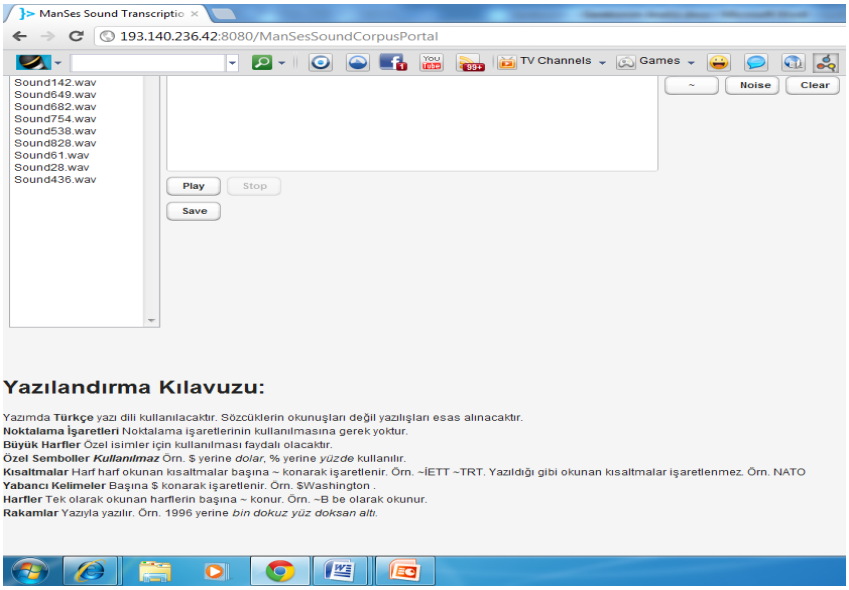
Daha büyük bir kullanıcı kitlesine ulaşıp hızlı ve verimli bir şekilde konuşma derlemi hazırlamak için web tabanlı bir platform geliştirildi. Makine öğrenimi gibi alanlarda kullanmak üzere etiketlenmiş veri toplamak için bu yöntem farklı çalışmalarda da kullanılıp Cowd-Sourcing olarak bilinmektedir [61, 62, 63]. Bu yöntemde, büyük bir

iş paketi küçük parçalar bölünüp daha büyük bir kullanıcı kitlesine ulaştırılıyor. Resim etiketleme, konuşmaların yazıya dökülmesi, metin çevirisi (translation) gibi birçok alanda bu yöntem kullanılmaktadır [64, 65]. Bu platformu sunan ve iş paketlerini dünyadaki çok sayıda insana ulaştıran en büyük servislerden birisi Amazon Mechanical Turk (AMT) servisidir. Bu platfrom üzerinde tüm dünyadan ve farklı alanlarda, insanlar tarafından yüklenen iş paketleri bulunup kullanıcılar tarafından işlenmektedir.

Crowd-Sourcing yönteminin hızlı ve ucuz olmasının yansıra bazı dezavantajları da var. Bu platform üzerinden yaptırılan işlerin kalitesi farklı yöntemler kullanarak kontrol edilip yapılan işin kalitesi ve doğruluğu kontrol edilmelidir. Çünkü bu platformda çalışan insanlar üzerinde kontrol mekanizması uygulamak daha zor olup ve bu da yapılan işin kalitesini doğrudan etkilemektedir. Bu sorunu engellemek ve elde edilen verilerin kalitesinden emin olmak için farklı mekanizmalar kullanılabilir. Aynı iş paketini birden fazla kişiye yaptırıp farklı kişiler tarafından aynı sonuç elde edildiği takdirde doğru olduğunu kabul edebiliriz. İş paketleri karşısında ödenecek olan ücret miktarını artırıp daha kaliteli bir iş yaptırmak ise ikinci bir yöntem olarak kullanılmaktadır. Diğer bir yöntem olarak da çeşitli işverenler tarafından onaylanan ve güvenirlik seviyesi yüksek olan insanlara iş yaptırıp yapılan işin kaliteli ve doğru olmasından emin olabiliriz [66, 67].

AMT gibi platformlarda dil bilgisi gerektirmeyen iş paketlerinin tamamlanması için kullanıcı bulmak daha kolaydır. Örneğin resim etiketleme işi dilden bağımsız olup tüm dünyadan çeşitli kullanıcılara yaptırılabilir. Ancak konuşmaların yazıya dökülmesi ve metin çevirisi gibi iş paketleri dile bağımlı olmalarından dolayı o dili bilen insanlar tarafından yapılması gerekmektedir. Türkçe bilen ve bu platform üzerinde çalışan kullanıcı sayısı oldukça az olup konuşma derlemi hazırlamak uzun sürmektedir. Dolayısıyla, daha hızlı ve verimli bir şekilde konuşma derlemi hazırlamak için bu tez kapsamında crowd-sourcing yöntemi uygulanıp yeni bir platfrom hazırlanmıştır. Web tabanlı bir portal hazırlayarak, farklı kullanıcıların bu portal üzerinden ses kayıtlarını dinleyip yazıya dökmelerine olanak sağlanmıştır. Bu servisin kullanıcıları daha çok Hacettepe Üniversitesi ve Başkent Üniversitesi öğrencelerinden oluşmaktadır. Şekil 3.1’de bu web uygulamasının ekran çıktısı verilmiştir. Bu uygulama üzerinde 100 farklı kullanıcı için hesap açılıp bunlardan 80 kullanıcı aktif olarak konuşma derlemi hazırlama sürecine katkıda bulunmuşlar.

Uygulamaya giriş yapan kullanıcı karşısına ses veri tabanından rastgele seçilen on farklı ses dosyası çıkıyor. Veri tabanında bulunan sesler YouTube'dan indirilen ve çeşitli konulardaki Türkçe konuşmalardan oluşmaktadır. İndirilen videolar genelde haber programları, ders anlatım videoları ve miting konuşmaları gibi videolardan oluşmaktadır. Videolardaki sesi görüntüden ayırdıktan sonra 16 Khz 16 bit PCM formatına dönüştürülmüştür. Daha sonra, ses dosyaları dinlenip gürültülü olan veya anlaşılması zor olan dosyalar işlemden çıkartılmıştır.



Şekil 3.1 Derlem hazırlamak için kullanılan Web uygulaması.

Ses dosyalarını daha küçük ses dosyalarına bölmek için ses aktivasyon tespiti (voice activity detection – VAD) algoritması [48] kullanarak sesteki boşluklara göre bölütlenmiştir. Bu süreç sonucunda elde edilen ses dosyalarının ortalama süresi 3.3 saniye olup kullanıcının dinlemesi ve konsantrasyonu açısından daha uygundur.

Konuşmalardaki yazıların standart ve doğru olması açısından bazı kurallar tanımlanıp ve uygulamayı kullanan kullanıcılardan bu kurallara uymaları istenmiştir. Yazı kuralları LDC HUB4 [68] projesi kapsamında hazırlanan ve konuşma derlemleri için standart olarak kullanılan kılavuza göre uyarlanmıştır. Uyulması istenilen yazım kuralları aşağıdaki gibidir:

- **Noktalama işaretleri:** Cümle sonunu belirten nokta (.), soru işareti (?), ünlem (!) ve özel isimlerden sonra gelen kesme (’) haricindeki noktalama işaretlerinin kullanılmasına gerek yoktur.
- **Büyük harfler:** Özel isimler için kullanımı faydalı olacaktır
- **Tam söylenmemiş sözcükler:** Başı ya da sonu duyulmayan kelimeler için (-) kullanılır. Örneğin, -zartesi, pazarte-. Bu durumda yazıcı duyulmayan kısımda ne söylendiğinden eminse bunu parantez içinde belirtebilir. Örneğin, (pa-)zartesi, pazarte(-si)
- **Özel semboller:** $, % gibi sembollerin kullanılmaması gerekiyor ve bunların yerine dolar, yüzde gibi yazılı halleri yazılmalıdır.
- **Kısaltmalar:** Harf harf okunan kısaltmaların başına ~ işareti konulur. Örneğin, ~TRT ve ~İETT . Yazıldığı gibi okunan kısaltmaların işaretlenmesine gerek yoktur. Örneğin NATO
- **Yabancı kelimeler:** Yabancı kelimeler $ ile işaretlenir. Örneğin $Washington. Yabancı dilde harf harf yazılan kısaltmalar $~ ile işaretlenir. Örneğin $~CIA
- **Rakamlar:** Rakamları yazı olarak yazılır. Örneğin bin dokuz yüz seksen altı
- **Duraksamalar ve ara sözler:** Bu sözler konuşma arasında kullanılır ve % ile işaretlenir. Örneğin, %hıhı (evet anlamında), %ıii, %aaa, %ooo

Kullanıcılar tarafından yazıya dökülen konuşmaların doğruluğunu kontro etmek için iki farklı kontrol mekanizması kullanılmıştır:

Çapraz iş paketi: Bu yöntemde aynı ses dosyası iki farklı kullanıcı tarafından dinlenip yazıya dökülüyor. İki kullanıcının da yazdığı metinler aynıysa o metin doğru olarak kabul edilip sistem tarafından konuşma derlemine eklenmektedir.

Güvenirlik değeri: Bu yöntemde, uygulamaya yüklenen ses kayıtları konuşma tanıma motorundan geçirilip elde edilen çıktı kullanıcı sonucu ile karşılaştırılıyor. Sonuçlar arasında %90 üstünde benzerlik varsa sonuç doğru kabul edilip derleme ekleniyor.

Web uygulamasını kullanarak on saatlik bir konuşma derlemi toplanmıştır. Sisteme yüklenen kayıtlar rastgele ve internetten alındığı için konuşmacı sayısı ve ortam türü gibi bilgiler bu derlem için bulunmamaktadır.

3.2. Veri Sentezleme

Makine öğrenimi alanında model eğitimi için kullanılan verinin niteliği ve niceliği bu modellerin başarısı açısından oldukça önem taşımaktadır. Yeterince iyi bir genelleme (generalization) yapabilen modeller yeterince çeşitli ve büyük veri kümesi kullanarak eğitilmektedir. Özellikle, sinir ağlarını model olarak kullanan alanlarda veri kümesinin büyük olması bu modellerin insan başarısına yakın sonuçlar üretmesine yol açmıştır [54]. DNN'lerde eğitilmesi gereken milyonlar veya milyarlarca parametre bulunuyor. Bu parametrelerin doğru hesaplanması için çeşitli veriler ve gerçek dünyayı temsil eden örneklere ihtiyaç duyulmaktadır. Konuşma tanıma probleminde ise farklı ortamlarda, farklı mikrofonlar ve çeşitli gürültü seviyesinde kaydedilen konuşmaların eğitim kümesinde bulunması bu sistemlerin gerekli performansı göstermesi için önemlidir.

Sesin kalitesi ve ortamdaki gürültü seviyesi konuşma tanımadaki kelime hata oranını etkiliyor. Hatanın kaynağı da genelde eğitim verisinin dağılımı ve özellikleri ile deney zamanındaki kullanılan veri arasındaki uyumsuzluktan kaynaklanıyor. Eğitim verisi gürültüsüz ve temiz kayıtları içerip tek bir mikrofondan elde edilmişse test kayıtlarındaki farklı ses kayıtları üzerindeki hata oranı yüksek çıkmaktadır. Bu sorunu aşmak için ses üzerinde gürültü düşürme (noise reduction) gibi birtakım ön işlemeler (pre-processing) yapılabilir. Ancak, bu yöntemler hem verinin içerdiği bilgi miktarını düşürüp hem de farklı ortamlar için farklı ön işlemelerin yapılması gerekiyor. Dolayısıyla, ses sinyali üzerinde ön işleme yapmak yerine kullanılan modelleri bu tür ortamlarda da başarı sağlayabilecek şekilde eğitmek daha mantıklı ve maliyeti düşük bir yöntemdir.

Son zamanlarda, farklı sinir ağları çeşitleri konuşma tanıma sistemlerinin tasarım ve gerçekleştirmesinde kullanılmaya başlamıştır [7]. Bu modeller, hem akustik modelleme hem de dil modeli oluşturmada kullanılıp klasik yöntemlere göre daha başarılı sonuçlar veriyor. Akustik modelleme tarafında GMM yerine sinir ağları olasılık dağılımı fonksiyonu olarak kullanılmaktadır. Dil modeli tarafında ise istatistiksel n-gram yerine sinir ağları kullanılmaktadır. Ancak, bu modellerde bulunan parametrelerin yeterince doğru hesaplanması için büyük miktarda konuşma ve metin verisi gerekmektedir.

Veri sentezleme gibi yöntemleri kullanarak eğitim setindeki ses kayıtları çeşitlendirip gerçek ortamda bulunan verilerin dağılımına yaklaştırılabilir. Örneğin, resim işlemede eğitim setindeki verileri gürültü ekleme, döndürme ve resmi farklı açılardan çekerek veri çeşitliliği artırılıyor [69]. Aynı yöntem konuşma tanıma modellerinin eğitimi için gerekli olan konuşma verileri üzerinde de yapılabilir [70, 71].

Konuşmalara çeşitli gürültü türleri ekleyip eğitim verisine dahil ederek akustik modeli bu tür ortamlara yönelik dayanıklı hale getirebiliriz. Diğer bir yöntem olarak da eğitim için kullanılan seslerin şiddeti (volume perturbing) ile oynayarak daha yavaş duyulan konuşmalar veya yüksek seste konuşulan sesleri de modelleyebiliriz. Üçüncü yöntem olarak ses dosyalarını hızını değiştirip (speed perturbing) farklı hızlarda konuşulan sesler ise eğitim kümesine dahil edilebilir. Bu sayede daha hızlı veya tane tane konuşulan konuşmalar konuşma tanıma sistemi tarafından daha yüksek bir başarı oranı ile tanınacaktır. Bu tezde, üç yöntemi de kullanarak eğitim seti olarak kullanılan konuşma derlemindeki verilerin çeşitliliği artırılıp sistemin hata payı düşürülmüştür.

3.2.1. Gürültü Ekleme

Gerçek bir konuşma tanıma senaryosunda ses sinyali farklı ortamlarda ve çeşitli cihazlarla kaydedilip konuşma tanıma servisine gönderiliyor. Dolayısıyla, bu çeşitliliği modellemek için benzer özellikleri taşıyan konuşma örneklerinin de eğitim verisi içerisinde bulunması gerekiyor. Özellikle taşınabilir cihazlardan (mobile devices) gönderilen konuşmalar farklı ortam gürültüleri ve sinyal aktarım kanalındaki özellikleri içerdikleri için bu seslerdeki başarı oranı daha düşüktür. Tüm bu özellikleri taşıyan konuşma örneklerinin hazırlanıp eğitim setine eklenmesi oldukça zaman alıcı ve pahalı bir süreçtir. Bu süreci kolaylaştırmak ve konuşma tanıma sisteminin başarısını artırmak için, eğitim setinde bulunan konuşmalara benzer gürültüler ekleyerek çeşitli ortamlardaki konuşmalar simüle edilebilir.

Çeşitli gürültü seslerinden oluşan bir havuz oluşturmak için üç farklı cep telefonu ile farklı ortamlardan gürültü kaydedilmiştir. Kaydedilen gürültüler arabada, alış veriş merkezinde, sokakta ve kafe gibi çeşitli ortamlarda kaydedilmiştir. Bu çalışma sonucunda, 16Khz ve 16 bit olarak kaydedilen yaklaşık iki saatlik bir gürültü havuzu oluşturulup eğitim derlemindeki konuşmaların farklı kısımlarına eklenmiştir.

Eğitilen modellerin bu gürültüleri öğrenip göz ardı etmemesi için rastgele olarak üç farklı gürültü dosyası seçilip sesin farklı kısımlarına eklenmiştir. Gürültü ekleme işleminde orijinal sesin uzunluğu değişmeyip sadece arka planda duyulan bir gürültü ekleniyor. Sentetik olarak eklenen bu seslerin şiddeti de her bir orijinal konuşma dosyası için rastgele seçilip 0.1 ile 0.5 değerleri arasındadır.

3.2.2. Hız Değiştirme

Konuşma dosyalarının hızı üzerinde oynama yaparak eğitim kümesindeki konuşmaların hız açısından çeşitliliği artırılabilir. Bu yöntemde, konuşmaların hızı 0.9 ve 1.1 ile çarpılıp bir konuşmanın daha yavaş ve daha hızlı versiyonları elde edildikten sonra eğitim kümesine ekleniyor. Dolayısıyla, farklı insanlar arasındaki konuşma hızı bu sesleri kullanarak modellenip konuşma tanıma başarısı arttırılabilir.

3.2.3. Ses Seviyesi

Konuşma tanıma başarısı farklı ses seviyelerinde kaydedilen konuşmalar için değişiklik göstermektedir. Özellikle, uzak mesafeden alınan ve şiddeti (volume) az olan seslerde tanıma oranı düşük olup veya hiç tanınmamaktadır. Ayrıca, yüksek bir ses seviyesindeki konuşma sinyalleri de sistem tarafından ilgisiz kelimeler olarak tanınıp veya gürültü olarak göz ardı edilebiliyor. Fakat gerçek bir kullanım alanında konuşmacının mikrofona olan mesafesiyle orantılı olarak konuşmadaki ses seviyesi daha yüksek veya daha düşük olabilir. Farklı ses seviyelerindeki tanıma oranını yüksek tutmak için konuşmalardaki ses seviyesini değiştirip eğitim derlemine eklenmiştir. Sox uygulamasını kullanarak orijinal konuşma derlemindeki her bir dosyasının ses seviyesi 0.9 ve 1.1 değerleri ile çarpılıp tekrar konuşma derlemine eklendi.

Bu tezde eğitim ve deney seti olarak kullandığımız veri kümeleri ile ilgili bilgiler Çizelge 3.2’de verilmiştir.

Çizelge 3.2 Eğitim, test ve doğrulama kümesindeki konuşma derlemlerinin bilgisi.

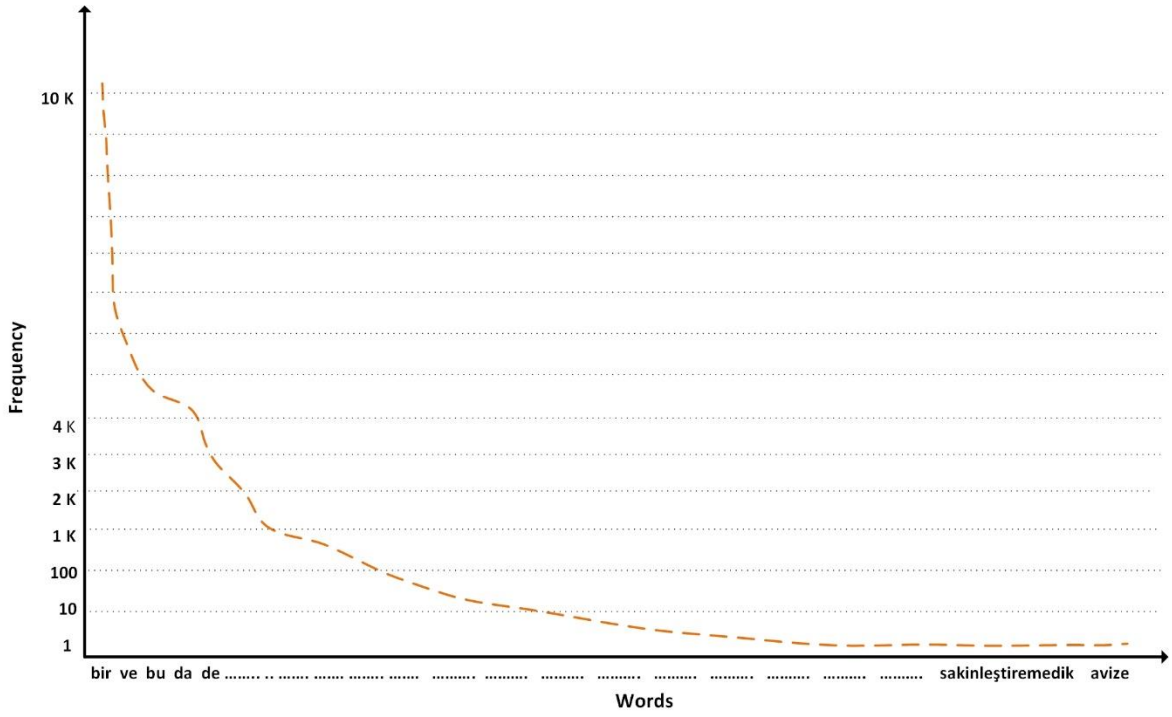
Source	Orginal	Speed Perturbing	Volume Perturbing	Noise Perturbing	Train	Test	Validation
TBN **[2]**	86	86	86	86	336	4	4
Mobil	4	4	4	4	14	1	1
Web	10	10	10	10	32	2	2
Total					**382**	**7**	**7**

[2] çalışmasındaki verinin 86 saatlik bir kısmı eğitim verisi olarak ayrılıp hız, şiddet ve gürültü ekleme yapıldıktan sonra 336 saat eğitim verisi elde edilmiştir. Aynı yöntemlerle, mobil ve Web derlemleri de çoğaltılıp toplamda 382 saatlik bir eğitim verisi oluşturulmuştur. Model eğitimi aşamasında hiper parametrelerin ayarlanması ve eğitimin erken durdurulması için 7 saatlik bir doğrulama seti kullanılacaktır. Ayrıca, her model eğitimi sonunda da kelime hata oranını ölçüp modelin doğruluk oranını incelemek için ayrı bir 7 saatlik test kümesi ayrılmıştır.

Eğitim setindeki kelime, cümle ve tekil kelime sayısı Çizelge 3.3'te verilmiştir. Ayrıca eğitim verisindeki kelimelerin sıklığı da Şekil 3.2'de çizilmiştir. Bu şekilden de anlaşıldığı üzere kelimelerin sıklığı Zipf kuralına uymaktadır.

Çizelge 3.3 Eğitim setindeki kelimelerin istatistiği. Tüm veri setinde toplam 138 bin tekil kelime bulunup bunların büyük bir kısmı TBN [2] çalışmasından gelmektedir. Sayılar ses sentezleme sonrası elde edilen derlemden hesaplanmıştır.

Source(Augmented)	#Word	#Sentence	Vocab Size
TBN	1.5 M	400 K	122 K
Mobil	80 K	15 K	11 K
Web	80 K	15 K	11 K
Total	1.6 M	430 K	250 K



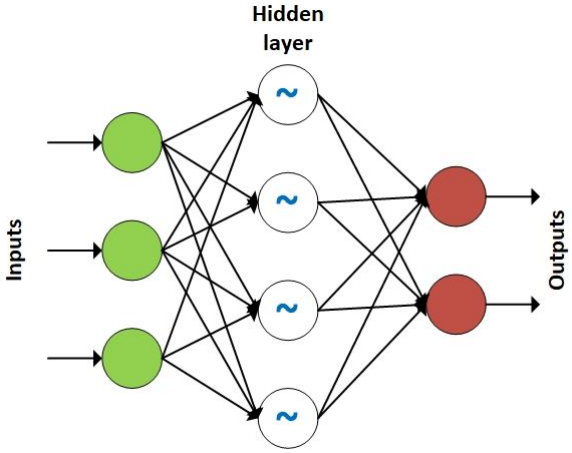
Şekil 3.2 Eğitim setindeki kelimelerin sıklığı. Kelimelerin dağılımı Zipf kuralına uymaktadır.

4. SİNİR AĞLARI

Yapay sinir ağları doğrusal bir yapısı olmayan problemlerde d_x boyutlu bir **x** girdisi ile d_y boyutlu **y** çıktısı arasındaki ilişkiyi modellemektedir. Bu yapı tarihsel olarak biyolojik sistemler tarafından incelenmiş olup insan beynindeki yapıdan esinlenmiştir [72]. İlk çalışmada bir algılayıcıyı (perceptron) doğrusal bir sınıflandırıcı olarak kullanıp matematiksel yapısı önerilmiştir [73]. Daha sonra, bu yapı doğrusal olmayan sınıflandırıcıların kullanımı için önerilip evrensel bir tahminleyici olarak (universal approximator) kullanılmaya başladı [74].

4.1. İleri Beslemeli Sinir Ağları

Sinir ağları farklı yapılara sahip olup çeşitli problemler için kullanılabilir. Bu modellerin en temel yapısı ileri beslemeli sinir ağlarıdır (feedforward neural network - FFNNET) ve diğer model tipleri bu yapı üzerine inşa edilmiştir. FFNNET yapı olarak hesaplama ünitelerinin farklı katmanlarda üst üste yığılmasında (stacked) elde edilmektedir. Hesaplama ünitesi literatürde nöron (neuron) olarak tanımlanıp kendisine gelen girdiler üzerinde birtakım matematiksel işlemler yapmaktadır. Şekil 4.1'de tek bir saklı katmandan (hidden layer) oluşan FFNNET modelinin grafiksel gösterimi verilmiştir. Bu şekilden de anlaşıldığı üzere, girdi olarak verilen bir vektör ara katmandaki nöronların matematiksel işlemlerine tabi tutulduktan sonra çıktı katmanına iletilmektedir. Çıktı katmanındaki üniteler ise problemin yapısına göre farklı anlamlara gelebilir. Örneğin, resim sınıflandırmada bu modelin çıktısı farklı objelerin olasılığını gösterebiliyorken, metinden duygu analizi yapan (sentiment analysis) bir modelde bu çıktılar metinin pozitif veya negatif olduğunu göstermektedir.



Şekil 4.1 Tek katmanlı ileri beslemeli sinir ağı.

Sinir ağı en basit mimaride, doğrusal bir katman, matris ve vektör çarpımları ve sonunda da doğrusal olmayan bir fonksiyon transferi ile ifade edilmektedir:

$$y = h(Mx + b) \tag{4.1}$$

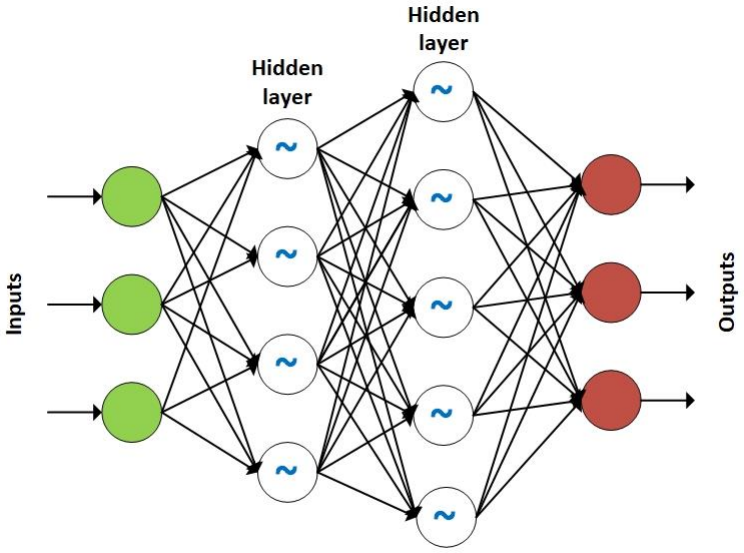
M matrisi $d_x \times d_y$ boyutlu ağrılık matrisi, b ise d_x boyutundaki bayas (bias) vektörünü gösterip modelin parametreleridir. h(.) fonksiyonu da doğrusal olmayan bir fonksiyonu ifade edip genelde *Tangent* veya *Sigmoid* fonksiyonları kullanılmaktadır.

Yapay sinir ağlarındaki en yaygın mimari genelde birden fazla katmanın birleşiminden oluşuyor. Bu yapı çok katmanlı algılayıcı (multi layer perceptron - MLP) olarak bilinir (Şekil 4.2). Tek katmandan oluşan bir MLP aşağıdaki gibi ifade edilir:

$$y_h = h(M_1 x + b_1) \tag{4.2}$$

$$y_{out} = h\big(f(x)\big) = h(M_2 y_h + b_2) \tag{4.3}$$

y_h saklı katmanın çıktısını gösterip f(x) fonksiyonu ise ağın çıktısını ifade ediyor. Ağırlık matrisleri ve bayas vektörleri modelin parametreleri olup eğitim aşamasında doğru değerleri bulunacaktır. Modeldeki saklı katman sayısı hiper parametredir ve modelin yapısı ve problemin karmaşıklığına göre deneysel olarak belirlenir.



Şekil 4.2 Çoklu katman sinir ağı.

Sinir ağları hem regresyon hem de sınıflandırma problemleri için kullanılabilir. Bu tezde, biz sinir ağlarını bir fonem sınıflandırıcısı olarak kullanıyoruz. Sınıflandırma probleminde, sinir ağının görevi **x** girdi vektörü ile etiketler i ∈ [1, . . . , I] arasındaki ilişkiyi modellemektir. Dolayısıyla, ağın çıktısı olan f(x) fonksiyonu, I boyutlu bir vektör olup $f_i(x)$ ise her bir sınıfın olasılığını göstermektedir. Sonsal olasılığı (posterior probability) hesaplamak için ağın çıktıları üzerinde *softmax* fonksiyonu uygulanmaktadır [75] :

$$P(i|x) = \frac{e^{f_i(x)}}{\sum_j e^{f_j(x)}} \tag{4.4}$$

Literatürde, çapraz entropi (cross-entropy) veya kare hata (squared-error) kriterlerini kullanarak sinir ağlarının yaptığı hata miktarı ölçülmektedir [76].

N tane eğitim örneği ve etiketi (x_n , i_n), *n=1,…,n* olan veri kümesini kullanarak, bir sinir ağı, L maliyet fonksiyonunu (objektif fonksiyonu veya kriter) optimize ederek eğitilmektedir. Genelde, ağın çıktısı ile doğru etiketler arasındaki farkı ölçen çapraz entropi fonksiyonu maliyet kriteri olarak kullanılıyor:

$$L(\theta) = \sum_{n=1}^{N} \log(P(i_n|x_n, \theta))$$

$$\log\big(P(i_n|x_n, \theta)\big) = f_i(x, \theta) - logadd(f_i(x, \theta)) \tag{4.5}$$

$$logadd\big(z_j\big) = \log(\sum_j e^{z_j})$$

Sinir ağındaki parametreli eğitmek için geri yayılım (back-propagation) algoritması kullanılmaktadır [77]. Bu algoritma çıktıdaki hatayı zincir türevleri kullanarak ağ üzerinde geriye dönük olarak yayıp modeldeki θ parametresi aşağıdaki denklemi kullanarak güncellenmektedir:

$$\theta \leftarrow \theta + \lambda \frac{\partial \log(P(i|x, \theta))}{\partial \theta} \tag{4.6}$$

λ değeri öğrenme oranı (learning rate) olup parametrelerin güncelleme miktarını kontrol ediyor. Parametrelerin (ağırlık ve bayas matrisleri) değeri rastgele olarak atanıp, kümeler (batch) üzerindeki maliyetin ortalaması veya rastgele seçilen örneklerdeki maliyetleri (stochatsic gradient descent - SGD) göz önünde bulundurarak güncellenebilir [78].

Sinir ağlarındaki en önemli problemlerden birisi aşırı uyma (overfitting) problemidir. Model sadece eğitim setindeki örnekleri çok iyi öğrenip deney setindeki örnekler üzerinde başarılı sonuçlar göstermeyerek iyi bir genelleme yapamıyor. Eğitim sırasında, doğrulama kümesi (validation set) kullanmak bu sorunu engellemek için kullanılan yöntemlerden birisidir. Her iterasyonda, modelin başarısı bu küme üzerinde ölçülüp doğruluk oranı düştüğü zaman eğitim erken durduruluyor [79]. Ayrıca, doğrulama kümesini hiper parametrelerin seçimi için de kullanılmaktadır.

4.2. Derin Sinir Ağları

İleri beslemeli modellerin başarısından sonra ve son yıllardaki hesaplama gücünün artışı ile birlikte derin sinir ağları kavramı ve modelleri kullanılmaya başladı. Bu tip ağlar birden fazla saklı katmanın üst üste eklenmesi sonucu ortaya çıkıyor:

$$y_{out} = h(M_n h(M_{n-1} \dots h(M_1 x))) \quad (4.7)$$

M_n , n'inci katmanın ağırlık matrisini gösterip h(.) ise aktivasyon fonksiyonudur. Bu yapı konuşma tanımadaki başarıyı tek bir katmandan oluşan MLP'lere göre daha da iyileştirdiği gözlemlenmiştir [80]. Ancak, eğitim verisi az olan durumlarda bu modellerde bulunan milyonlarca parametrenin doğru değerleri hesaplanamamaktadır [81]. Bu sorunu ortadan kaldırmak için daha önceden eğitilmiş modellerin kullanımı önerilmiştir. Bu yöntem üretken modelleri (generative model) kullanarak iyi bir ara öznitelik öğrenmeye dayanmaktadır. Daha sonra bu ara öznitelikleri kullanarak ayrımcı model eğitimleri yapılmaktadır.

Literatürde bulunan hibrid HMM/DNN tabanlı sistemler, sesteki spektral (cepstral) özellikleri fonem tanıma [82] ve spontane konuşma tanımadaki girdilerin öznitelik vektörü olarak kullanıyorlar. Tıkanık (bottleneck) öznitelik vektörlerini kullanarak konuşma tanıma yapan çalışmalar ise mevcuttur [80]. Aygıta atma (dropout) yöntemi ise aşırı uymayı engellemek için normalizasyon yöntemi olarak konuşma tanımada denenmiştir [51]. Yaygın olan öznitelik vektörlerini kullanmadan direk ham sinyalden öznitelik vektörlerini öğrenen ve tanımayı uçtan uca gerçekleştiren sinir ağları ile ilgili de çalışmalar bulunmaktadır [83, 84].

4.3. Konuşma Tanımada Derin Sinir Ağları

Günümüzdeki birçok konuşma tanıma sistemi HMM yapısını sesteki dinamikliği ve GMM modelini de HMM'deki her bir durumun (state) olasılık dağılımı fonksiyonu olarak kullanmaktadır. Durumlar içerisindeki GMM dağılımı sesteki bir pencerenin (frame) ne kadar o duruma uyduğunu (fit) gösterip birden fazla gauss modelinin karışımından oluşuyor. Yıllar içerisinde farklı algoritmaların ortaya çıkması ve hızlı donanımların kullanılabilmesi ve ucuzlaması ile birlikte HMM/GMM tabanlı konuşma tanıma uygulamalarının doğruluk oranı giderek artmıştır. Bu algoritmalardan en önemlisi de beklenti maksimizasyonu (expectation maximization - EM) algoritmasının [85, 86] HMM-GMM tabanlı akustik modelindeki parametrelerin bulunması için kullanılmasıdır. Bu sistemin akustik girdisi sesteki öznitelik vektörleri olup genelde MFCC veya PLP yöntemlerinin katsayıları kullanılmaktadır.

GMM'lerdeki güçlü modelleme kapasitesi ve bu modellerdeki parametrelerin hesaplanması için hızlı çalışabilen EM gibi algoritmaların çalışmasından dolayı bu modelleri konuşma tanıma problemi için doğru bir aday olarak kabul edebiliriz.

Üretken (generative) yöntemlerle model eğitildikten sonra ayrımcı eğitim (discriminative training) algoritmaları ile ince ayarlar (fine tuning) yapılırsa bu modellerin başarısı daha da arttırılabilir. Özelliklede, ayrımcı eğitim için kullanılan objektif fonksiyonu fonem veya kelime hata oranı olursa tanıma başarısındaki iyileşme daha da artmaktadır [10].

Bütün bu artılara rağmen GMM'ler doğrusal yapısı olmayan verilerin modellemesi için yetersiz kalıyor. Örneğin, küre yapısı olan bir veri kümesi birkaç parametresi olan basit bir modelle ifade edilebiliyorken GMM kullanıldığı takdirde yüzlerce parametresi olan bir modele ihtiyaç duyulmaktadır. Ancak, konuşma sinyalleri birkaç basit parametresi olan bir dinamik sistemden üretilmektedir. Dolayısıyla, sesteki büyük bir pencere içerisindeki bilgiler kullanarak daha basit bir model yapısıyla konuşmadaki yapı modellenebilmelidir.

Son yıllarda, donanımların hızlanması ve ucuzlaması ile birlikte sinir ağları eğitimi için kullanılan geri yayılım algoritması ile çok katmanlı modellerin eğitimi mantıklı bir zaman süresi içerisinde mümkün olmuştur. Özellikle, grafik işleme ünitesi (graphical processing unit - GPU) gibi binlerce hesaplama çekirdeği içeren donanımlar sayesinde büyük veri kümeleri kullanarak çok katmanlı ve geniş çıktı katmanı (output layer) olan modeller eğitilebiliyor. Son çalışmalarda, sinir ağları kullanan büyük sözlüklü konuşma tanıma sistemlerinin GMM tabanlı sistemlere göre daha başarılı sonuçlar ürettiği tespit edilip bu yapıyı kullanan akademik ve ticari uygulamalar da ortaya çıkmaya başlamıştır [1]

4.4. Model Eğitimi

Derin sinir ağı girdi ve çıktı katmanları arasında birden fazla saklı katman bulunan ileri beslemeli sinir ağıdır. Şekil 4.3'ten de anlaşıldığı üzere girdi vektörü bir DNN'deki girdi katmanından ağa giriş yapıp ara katmanlarda bulunan algılayıcılar (perceptron) gerekli işlemleri yaptıktan sonra son katmandaki üniteler bu veriye ait sınıfı belirliyor. Verilen bir girdiye ait doğru olan çıktının veya sınıfın elde edilmesi için modeldeki parametrelerin uygun değerleri bulunmalıdır. Bu değerler model eğitimi aşamasında objektif fonksiyonu minimize ederek sürekli güncellenip eğitim verisi üzerinde doğru çıktıları üreten parametre değerleri bulunuyor.

- **Model Parametreleri**

Matematiksel bir model, kendisinde bulunan ve hesaplanması gereken parametreleri ile eğitilip veri örneklerini sınıflandırabilir. Sinir ağları da evrensel bir hesaplama modeli olup bu modelde bulunan parametrelerin doğru değerleri bulunursa sınıflandırma veya regresyon gibi problemleri için kullanılabilir. Sinir ağındaki her katman bir **W** ağırlık matrisi ve **b** bayas vektörü ile ifade edilmektedir. Katmandaki her bir aygıt, kendisine gelen girdileri **W** matrisi ile ağırlandırıp daha sonra b vektöründeki değerler ile topluyor:

$$\mathbf{W}=\begin{bmatrix} w_{11} & \cdots & w_{1m} \\ \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots \\ w_{n1} & \cdots & w_{nm} \end{bmatrix} \quad \mathbf{b}=\begin{bmatrix} b_1 \\ \cdots \\ \cdots \\ b_n \end{bmatrix}$$

Model eğitirken bu parametrelerin değerleri SGD algoritması ile sürekli güncellenip doğru çıktıları elde edene kadar bu güncelleme devam ediyor.

Saklı katmanlarda bulunan her bir ünitenin görevi kendisinden bir önceki katmandan gelen $\mathbf{x_j}$ değerini logistic fonksiyonundan (logistic function) geçirip y_j sonucunu bir sonraki katmana göndermektir:

$$y_j = logistic(x_j) = \frac{1}{1+e^{-x_j}}, \qquad x_j = b_j + \sum_i y_i \, w_{ij} \tag{4.8}$$

b_j saklı katmanda bulunan j ünitesinin sapma değeri olup w_{ij} bir önceki i katmanından gelen değerlerin ağılığıdır. Birden fazla sınıf bulunan sınıflandırma probleminde son katmandaki ünitelerin değeri *Softmax* fonksiyonu ile normalize edilip sonuçlar olasılık olarak anlamlandırılabilir:

$$p_j = \frac{\exp(x_j)}{\sum_k \exp(x_k)} \tag{4.9}$$

DNN modelinde bulunan parametreler, maliyet fonksiyonunun (cost function) türevini geriye doğru yayıp (backpropagation) hata miktarını minimize ederek eğitilmektedir. Maliyet fonksiyonu aslında eğitim verilerine ait doğru sınıf ile modelin tahmin ettiği sınıf arasındaki tutarsızlık miktarıdır. Bu fonksiyon, *Softmax*

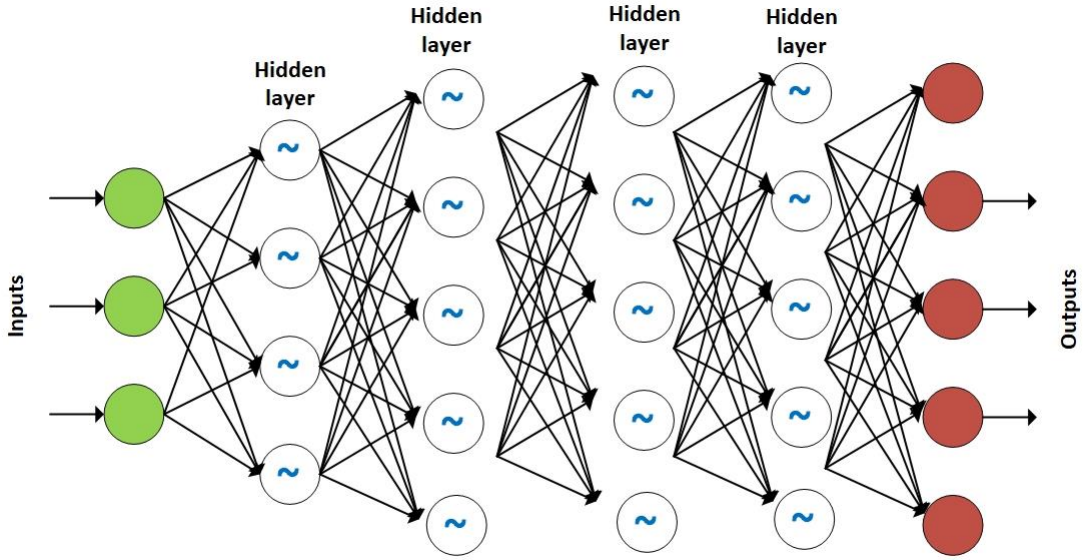
fonksiyonunun ürettiği p olasılığı ile doğru hedef sınıf olan d arasındaki çapraz entropi (cross entropy) miktarını ölçmektedir:

$$C = -\sum_{j} d_j \log p_j \tag{4.10}$$

DNN'ın son katmanında bulunan ünitelerin sayısı eğitim verilerinin atanması gereken sınıf sayısı kadardır. Eğitim sırasında her bir eğitim örneği için son katmandaki ünitelerden bir tanesinin değeri 1'e ve diğerlerinin değerini 0'a yakınlaştırmaya çalışılıyor. Bu şekilde gösterilen sınıflandırma yöntemi one-hot-vector olarak adlandırılıp çoklu sınıflandırma problemlerinde yaygın olarak kullanılmaktadır.

Büyük veri kümeleri üzerinde eğitim yapılırken DNN'deki türev hesaplamaları bu kümeden alınan daha küçük bir parça (mini batch training) üzerinde yapılmaktadır. Bu yöntemde, maliyet fonksiyonunun değeri daha küçük parçalar üzerinden hesaplanıp parametre güncellemeleri bir parçada bulunan tüm veri örneklerinden elde edilen hata miktarının ortalaması ile yapılmaktadır. Eğitim sırasında yapılan güncellemelerin daha küçük adımlar ve kontrollü yapılması için öğrenme oranı (learning rate) diye bir parametre alınan türevlerin güncellemelerdeki etkisini kontrol etmektedir:

$$\Delta w_{ij} = \alpha \Delta w_{ij}(t-1) - \epsilon \frac{\partial C}{\partial w_{ij}(t)} \tag{4.11}$$



Şekil 4.3 Derin sinir ağı birden fazla saklı katmanın üst üste yığılmasından elde edilmektedir.

- **Model Başlatma (Model Initialization)**

DNN'de bulunan **W** ve **b** parametrelerinin doğru değerleri eğitim sırasında hesaplanıyor. Ancak, eğitimin hızlı sonuçlanması ve optimizasyon algoritmalarının lokal minimumlara düşmemesi için bu matrislerin başlangıç değerleri doğru seçilmelidir. DNN'ler doğrusal modeller olmadıklarından dolayı ve eğitim kriterleri model parametrelerine göre konveks bir yapıya sahip olmadıkları için model eğitiminin doğru noktadan başlatılması önem taşımaktadır. Parametrelere başlangıç değeri atarken iki konunun göz önünde bulundurulması gerekir. Parametrelerin değeri aygıtlar içinde bulunan aktivasyon fonksiyonunun (activation function) doğrusal bir skalada çalışmasını sağlamalıdır.

Eğitim parametrelerine başlangıç değeri atamak için kullanılan yöntemlerden birisi [87] bu değerleri ortalaması sıfır ve standart sapması $\sigma = \frac{1}{\sqrt{N_l}}$ olan bir Gaussian dağılımından çekmektir. Standart sapmadaki N_l o katmandaki bir aygıta gelen girdi sayısıdır.

4.5. Aygıt Atmak

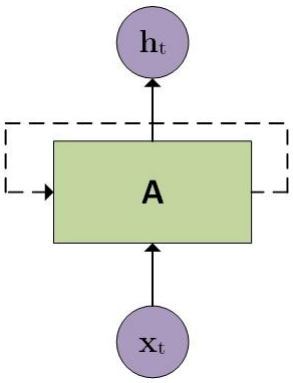
Aşırı uyma (over fitting) problemini önlemek için kullanılan yöntemlerden birisi aygıt atma (dropout) yöntemidir [88]. Bu yöntemde, eğitim sırasındaki her iterasyonda

katmanlardaki aygıtların belli bir kısmı (α kadar) rastgele olarak atılıp kendilerine gelen ağırlık değerleri sıfıra çekiliyor. Geri kalan (1- α kadar) aygıtlar gerekli olan hesaplamaları yapıp eğitim parametreleri tahmin ediliyor. Aygıtların bu şekilde atılması aslında eğitim verisindeki örneklere bir nevi gürültü ekleme demektir. Belli bir aygıt aynı veri için kendisinden önceki katmanda bulunan farklı aygıtlardan girdi alabilmektedir. Dolayısıyla, modelin hep aynı öznitelikler ile örüntüleri çıkarması engellenmiş olup aynı veri örneğini farklı öznitelikler ile tanıyabilmekteyiz.

4.6. Tekrarlanan Sinir Ağları

İnsanlar her saniye sıfırdan ve geçmişi hatırlamadan düşünmeye başlamıyorlar. Bir metini okurken, her kelimeyi bir önceki kelimeleri ve cümleleri anlayıp hatırlayarak devam ediyoruz. Bu şekilde sürekli geçmişteki öğrendiklerimiz ve bilgilerimizi hafızamızda tutup yeni şeyler öğrenerek bilgimizi arttırıyoruz.

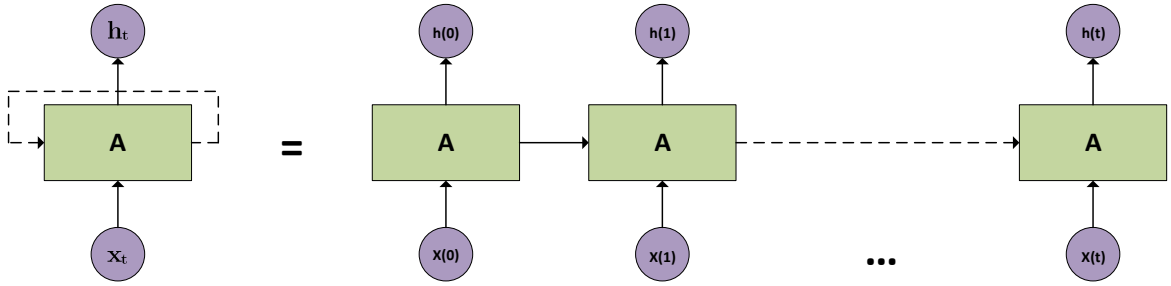
Geleneksel sinir ağlarında geçmişteki bilgileri hatırlama mekanizması bulunmamaktadır. Örneğin, bir filimdeki olayları takip edip ilerleyen dakikalarda ne olacağını tahmin etmek istiyorsak ileri beslemeli bir sinir ağı ile bunu yapamayız. Tekrarlanan sinir ağları (recurrent neural network - RNN) bilginin farklı adımlarda aktarılmasını sağlamak için tasarlanıp zaman içerisindeki ve dizi halinde olan verilerin işlenmesi için tasarlanmıştır (Şekil 4.4).



Şekil 4.4 Bir önceki adımdan elde edilen çıktıyı girdi olarak kullanan RNN yapısı.

Bir sinir ağı modeli olan **A** ünitesi **t** zamanındaki $\mathbf{x}_t$ girdisini alıp $\mathbf{h}_t$ çıktısını üretiyor. RNN'lerdeki döngü yapısı, üretilen çıktını t+1 zamanında girdi olarak kullanılabilmesini sağlamaktadır. Bu yapı, zaman serisi halinde olan ve zaman adımları arasında ilişki bulunan verilerin işlenmesini bir sinir ağı ile mümkün kılmaktadır.

RNN modeli, birden fazla sinir ağının zaman içerisindeki tekrarlanması ile elde edilmektedir. Her bir ünite o andaki veriyi işleyip elde ettiği çıktıyı bir sonraki adıma iletiyor. Şekil 4.5'ten de anlaşıldığı üzere bu yapı seri halindeki verilerin işlenmesi için kullanılabilir.



Şekil 4.5 RNN'nin zaman adımları içerisinde tekrarlanması.

Son yıllarda RNN'ler farklı problemlerin çözümü için kullanılmaya başladı. RNN tabanlı dil modelleri uzun bir sıradaki kelime bağımlılıklarını modelleyebildikleri için *n-gram* yöntemine göre daha başarılı sonuçlar veriyor [89, 90, 91]. Makine çevirisi [92, 93, 94, 95] , doğal dil işleme, resim işleme [96, 97, 98, 99, 100] ve anlama gibi alanlarda ise RNN'ler oldukça başarılı sonuçlar vermiştir. Konuşma ve konuşmacı tanıma/doğrulama alanlarında da bu tip modellerin kullanımı oldukça yaygınlaşıp HMM tabanlı sistemler göre daha başarılı sonuçlar elde edilmiştir [7, 57, 70, 71, 101, 102].

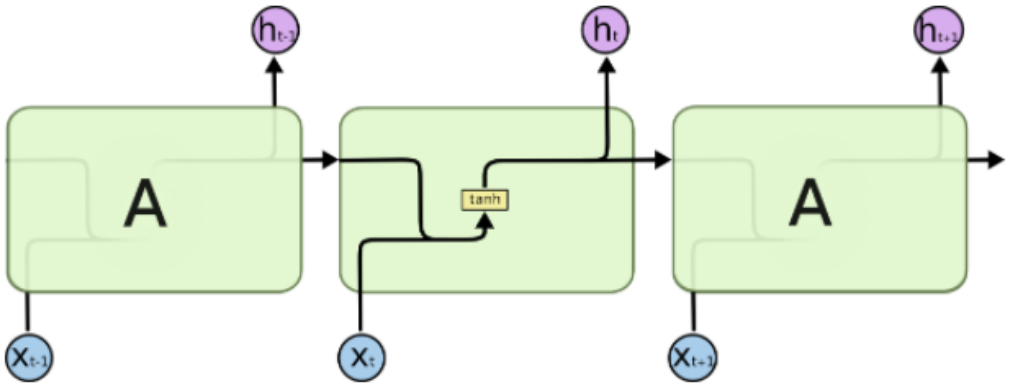
4.6.1. RNN'lerdeki Unutkanlık Problemi

RNN'lerin getirdiği en büyük avantaj daha önce görülen girdilerin bilgisini bir sonraki adımlara taşıyabilmektir. Bazen bir girdi serisinde sadece birkaç adım öncenin bilgisine ihtiyacımız var. Örneğin, "bugün güneş var ve --- çok sıcak" cümlesindeki kelimeyi bir dil modelinin tahmin etmesi için sadece geçmişteki birkaç kelimeyi hatırlaması yeterlidir. Bu durumlarda RNN'ler uzun bir geçmiş ile karşı karşıya olmadıkları için bağımlılıkları iyi modelleyebiliyorlar. Ancak, "ben Türkiye'de yaşıyorum ve yıllardır ... ***Türkçe'yi*** de iyi konuşabiliyorum " cümlesindeki "*Türkçe'yi"* kelimesinin model tarafından tahmin edilebilmesi için 10-15 kelime öncesindeki "Türkiye'de" kelimesinin model tarafından hatırlanması gerekiyor. Dolayısıyla bu tür kullanımlarda modelin uzun bir geçmiş hakkında bilgi sahibi olup ve bu bilgileri unutmaması gerkmektedir.

İlk girdi ile son girdi arasındaki mesafe uzadıkça RNN'ler geçmişte gördükleri bilgileri unutmaya başlıyorlar. Teorik olarak, RNN'ler bu tür uzun bağımlılıkları hatırlayıp unutmamaları gerekmekte, fakat gerçek kullanımlarda bu her zaman doğru olmayıp model uzun girdilerde unutkan olmaya başlıyor [103, 104]. Bu sorunu çözmek için Uzun Kısa Süreli Bellek (long short-term memory - LSTM) üniteleri önerildi [103, 105]. Bu yapılar, uzun vadeli bağımlılıkları hatırlamaya ve dolayısıyla "bağlam" farkındalığına sahip sinir ağları elde etmek için kullanılmaktadır.

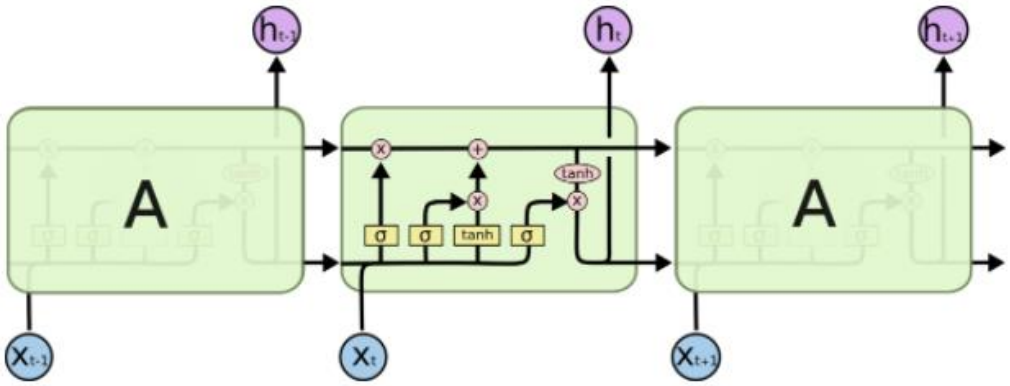
4.6.2. Uzun Kısa-Süreli Bellek – LSTM

LSTM'ler, RNN modelinde kullanılan ve uzun bağımlılıkların model tarafından hatırlanmasına yardımcı olan ünitelerdir [103, 105]. Standart bir RNN'de basit bir şekilde sinir ağının çıktısı bir sonraki girdi ile birlikte aktivasyon fonksiyonuna tabi tutuluyor (Şekil 4.6)



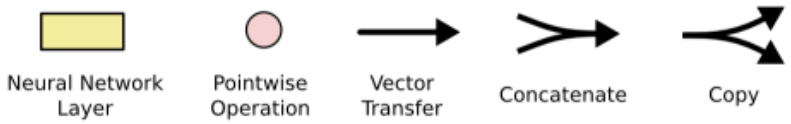
Şekil 4.6 RNN'de bir girdi ile katmanın önceki çıktısı birlikte kullanılıyor.

LSTM'ler de aynı tekrarlamayı yapıyorlar; ancak RNN içerisindeki yapı biraz daha farklıdır. Tek bir katman yerine dört farklı katman özel bir biçimde çalışıp bellek yapısını modelliyorlar (Şekil 4.7).



Şekil 4.7 LSTM ünitesi içerisindeki bileşenler.

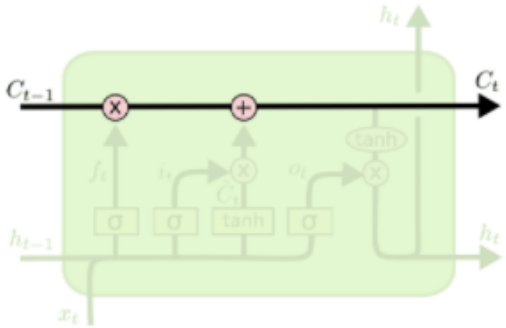
Şekil 4.7'deki bileşenlerin anlamı Şekil 4.8'de verilmiştir.



Şekil 4.8 LSTM ünitesi içerisindeki sembollerin açıklaması.

Şekil 4.7'deki her bir siyah çizgi bir düğümün (node) çıktısını başka bir düğüme taşıyor. Pembe daireler ise vektör toplaması gibi nokta tabanlı (pointwise) işlemleri gösterip, sarı dikdörtgenler de eğitilmesi gereken katmanları ifade etmekteler. İki çizgini birleşmesi vektörlerin *merge* edilmesi anlamına gelip ikiye ayrılan çizgi ise aynı bilgini kopyalanması anlamına gelmektedir.

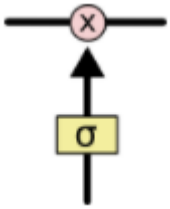
LSTM'deki çekirdek kavram ünitenin durumudur (cell state) ve LSTM ünitesini üst kısmından geçen siyah çizgi ile Şekil 4.9'da gösterilmiştir.



Şekil 4.9 LSTM ünitesinin durumunu taşıyan siyah çizgi .

LSTM'in durumu bir taşıma köprüsü görevini yapıp tekrarlanma boyunca tüm zinciri birbirine bağlı tutulmasını sağlar. Ünitenin durumu farklı geçitler (gate) ile güncelleniyor ve gerektiği zaman bilgiler ilave edilip veya çıkartılabilir.

Geçitler, *sigmoid* fonksiyonu kullanan sinir ağı katmanı olup geçirilmesi gereken bilgi miktarını kontrol ediyor (Şekil 4.10).



Şekil 4.10 LSTM ünitesindeki bir geçit.

Sigmoid bir geçidin çıktı değerleri 0 ile 1 arası olduğu için yukarıdan geçen bilgi ile çarpıldığında bu bilginin hangi miktarda iletilmesi gerektiğini belirlemiş oluyor. Bu değer 0 olduğu zaman bilgi iletilmeyip 1 olduğu zaman da gelen vektör olduğu gibi iletiliyor.

Özet olarak LSTM'in gerçekleştirdiği adımlar Çizelge 4.1'de açıklanmıştır. İlk adımda durum içerisinde hangi bilgilerin geçirilmesi gerektiğine karar veriliyor. Bu adım unutma geçidi olan (forget gate) bir sigmoid katmanı ile gerçekleştiriliyor. İkinci adımda, ünitenin durumuna hangi yeni bilgilerin eklenmesi gerektiği ile ilgili karar veriliyor. Bir sonraki adımda da, ünite durumu güncelleniyor ve yeni durum olarak kaydediliyor. Son adımda da, ünitenin saklı vektörü olan h_t ve durum bilgisini içeren c_t çıktıları üretilip bir sonraki üniteye iletiliyor.

LSTM yapısı RNN'lerin unutkanlık sorununu çözüp uzun dizilerdeki bilginin kaybolmamasını sağlıyor. LSTM'lere benzer görevi yapan farklı ünitelerde

bulunmaktadır. Geçitli Tekrarlana Üniteler (gated recurrent unit – GRU) LSTM yapısına benzer bir yapısı olup daha az parametre içermektedir. Bu ünitelerde son zamanlarda farklı problemlerde kullanılmaya başladı ve daha az parametre içerdiği için LSTM'e göre eğitim süresi daha kısadır.

4.7. Diziden Diziye Modeller

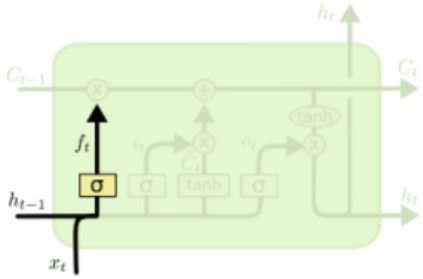
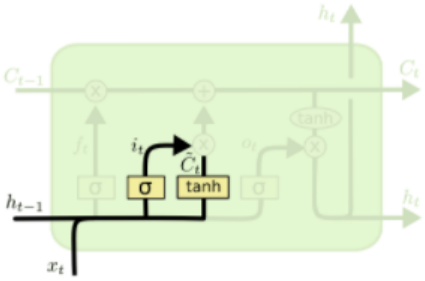
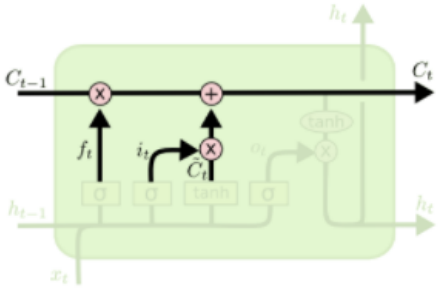
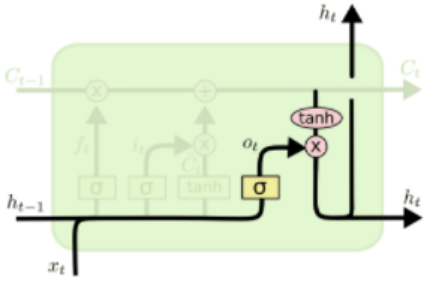
Diziden diziye modeller (sequence-to-sequence – seq2seq) makine çevirisi, konuşma tanıma ve metin özetleme gibi bir dizinin başka diziye çevrilmesi gereken problemlerde kullanılmaktadır. Bu başlıkta bu model yapısı incelenip dizi halinde olan verilerde nasıl kullanılabileceği ile ilgili örnekler verilecektir.

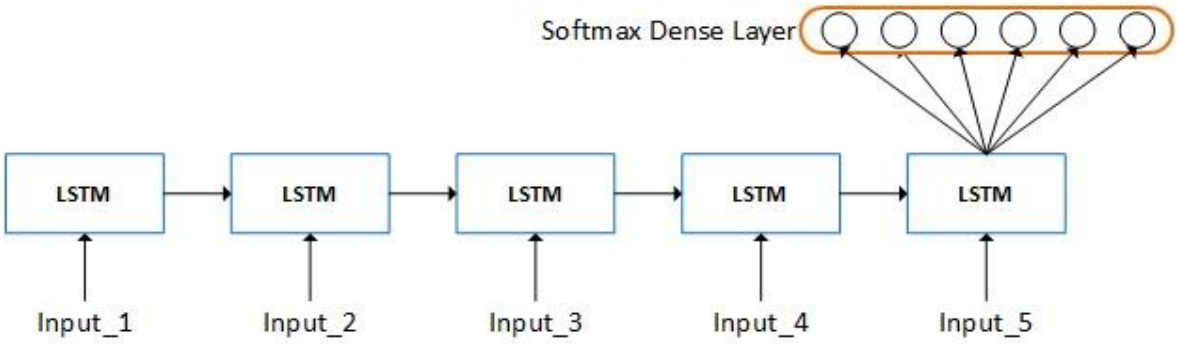
4.7.1. Dizi Sınıflandırma

RNN'ler dizi halindeki bir girdiyi sırayla alıp önceki bilgiyi de hatırlayarak girdi vektörlerini sonuna kadar işleyebiliyorlar [106]. Bu süreç kodlama aşaması olup kodlayıcının (encoder) girdi dizisini işleyerek gerçekleştiriliyor. Son girdi vektörü de işlendikten sonra elde edilen bilgi vektörü (thought vector) üzerinde bir sınıflandırma yapılabilir. Problemdeki sınıf sayısına göre bilgi vektörü bir saklı katmandan geçirilip çıktılar üzerinde *softmax* fonksiyonu uygulanırsa her bir sınıfın olasılığı elde edilir (Şekil 4.11).

Örneğin, duygu analizi yapan bir RNN'de cümledeki kelimeler kodlayıcı ile kodlanıp daha sonra pozitif/negatif/nötr olarak sınıflandırılabilir. Konuşmacı tanıma/doğrulama uygulamasında ses sinyaline ait pencereler sırayla kodlanıp daha sonra kişi tespiti veya kimlik doğrulaması yapılabilir.

Çizelge 4.1 LSTM ünitesi içerisinde gerçekleşen adımlar ve yapılan hesaplamalar [107].

Adım	Gösterim	Hesaplama
Ünite durumundaki geçirilmesi gereken bilgi miktarı		$f_t = \sigma(w_f.[h_{t-1}, x_t] + b_f)$
Ünite durumunda hangi yeni bilgilerin yerleştirilmesine karar vermek		$i_t = \sigma(w_i.[h_{t-1}, x_t] + b_i)$ $C_t = \sigma(\tanh(w_C.[h_{t-1}, x_t] + b_C))$
Eski durum bilgisinin güncellenmesi		$C_t = f_t * C_{t-1} + i_t * C_t$
Son olarak çıktıların üretilmesi		$o_t = \sigma(w_o[h_{t-1}, x_t] + b_o)$ $h_t = o_t * \tanh(C_t)$

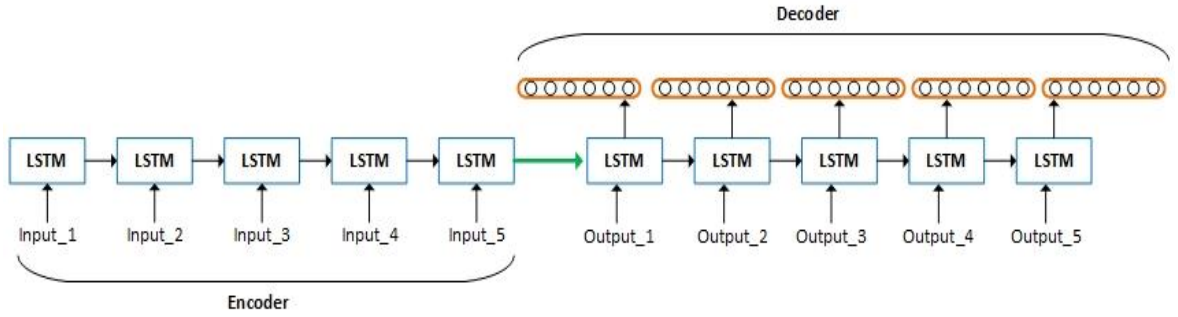


Şekil 4.11 RNN ile Dizi sınıflandırma. Girdiler sırayla işlenip en son bir *softmax* katmanı ile sınıflandırılıyorlar.

4.7.2. Diziden Diziye

Çoğu makine öğrenimi probleminde girdideki bir diziyi başka bir diziye dönüştürmek istiyoruz. Konuşma tanımadaki amacımız da sinyal dizisini başka bir karakter dizisine çevirerek tanımayı gerçekleştirmektir. Makine çevirisi probleminde, bir dildeki kelime dizisini başka dildeki kelime dizisine çevirmek istiyoruz [93]. Bu tür problemlerin hepsinin ortak özelliği bir girdi dizisini başka bir diziye çevirmek olup RNN yapısı ile modellenebilir.

Dizi sınıflandırma bölümünde anlatılan kodlayıcı taraf son girdiye de işledikten sonra elde edilen bilgi vektörünü RNN tipinden olan bir çözücüye (decoder) iletip dizi halindeki çıktılar üretilebilir. Bu yapı, kodlayıcı-çözücü (encoder-decoder) mimarisine sahip olup bir diziyi başka diziye dönüştürmek için kullanılabilir (Şekil 4.12).



Şekil 4.12 Bir diziyi başka bir diziye çeviren RNN modeli. Kodlayıcı, girdi dizisini işleyip çıktıların üretilmesi üzere bilgi vektörünü çözücü tarafa iletiyor.

Modelin ürettiği çıktılarla gerçek (ground-truth) çıktılar arasındaki fark ölçülüp çapraz entropi (cross-entropy) fonksiyonu objektif fonksiyon olarak kullanılabilir. Stochastic Gradient Descent (SGD) ve Adam Optimizer gibi çeşitli optimizasyon algoritmaları [108, 109] ile model parametreleri güncellenip objektif fonksiyonu minimizde edildiği zaman bir diziyi başka diziye çevirebilen RNN modeli eğitilmiş olur.

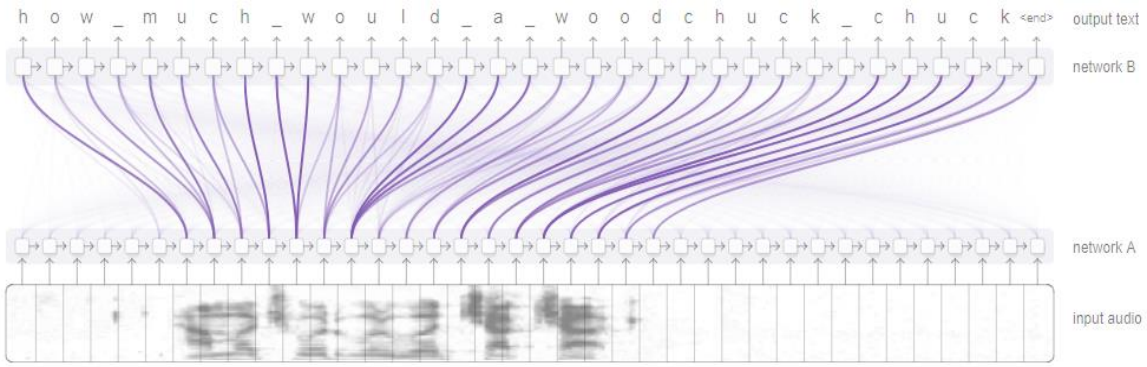
4.8. Odaklanma Mekanizması

Kodlayıcı - Çözücü (encoder - decoder) yapısı oldukça popüler bir yapı olup makine öğrenimi ile ilgili çeşitli problemler için kullanılabilir. Bu modelde, kodlayıcı taraf uzun bir girdi dizisini kodlayıp bilgi vektörünü üretiyor. Ancak tüm bu girdilerdeki bilgilerin tek bir vektör içerisine yerleştirilmesi bilgi kaybına neden olabiliyor. Bu da girdi dizisindeki uzunluğa bir kısıtlama getirmiş olup uzun girdiler için modelin performansı düşürmektedir. Bu kısıtlamayı RNN'lerden kaldırmak için odaklanma mekanizmasını (attention mechanism) bu bölümde inceleyeceğiz.

Odaklanma mekanizması insanın gözündeki ve beynindeki odaklanma yapısından esinlenmiştir. İnsan bir objeye baktığı zaman o objeyi daha net ve yüksek çözünürlük ile görüp çevresindekileri bulanık görmektedir (Şekil 4.13) [110].

Şekil 4.13 İnsan bir objeye odaklandığı zaman o objeyi daha net görüp çevresindekileri bulanık görüyor [110].

İnsan beynindeki bu özelliği RNN yapısında da kullanabilirsek bir çıktıyı üretmek için girdi dizisindeki hangi sembollere daha fazla odaklanmamız gerektiğini modele öğretebiliriz. Örneğin, makine çevirisinde bir kelimeyi çevirmek için o kelimenin kendisine ile iki sağ/solundaki kelimelere odaklanmamız önemli olabilir. Konuşma tanımda da bir karakterin üretilmesi için girdi tarafındaki o karaktere ait sinyal çerçevelerine daha fazla ağırlık verip ilgili çerçevelerden uzaklaştıkça odaklanmayı azaltabiliriz. Model eğitimi aşamasında, çözücü hem doğru çıktıları üretmesini hem de odaklanması gereken girdileri öğreniyor.



Şekil 4.14 Karakter dizisini üretmek için hangi ses çerçevelerine daha fazla odaklanması gerektiğini odaklanma mekanizması belirliyor. Daha koyu olan çizgiler ilgili çerçevelerin ilgili karakteri üretmek için daha önemli olduğunu gösteriyor [111].

Odaklanma mekanizması farklı yöntemlerle uygulanabilir. Bunlardan ünlü olanı ve makine çevirisinde kullanılan [112] yöntemin adımları aşağıdaki gibidir:

- Girdideki her bir LSTM ünitesinin saklı durumu (hidden state) çıktılardaki tüm ünitelerin saklı durumu ile karşılaştırılıp odaklanma ağrılıkları (attention weights) hesaplanıyor (Şekil 4.15 adım 1):

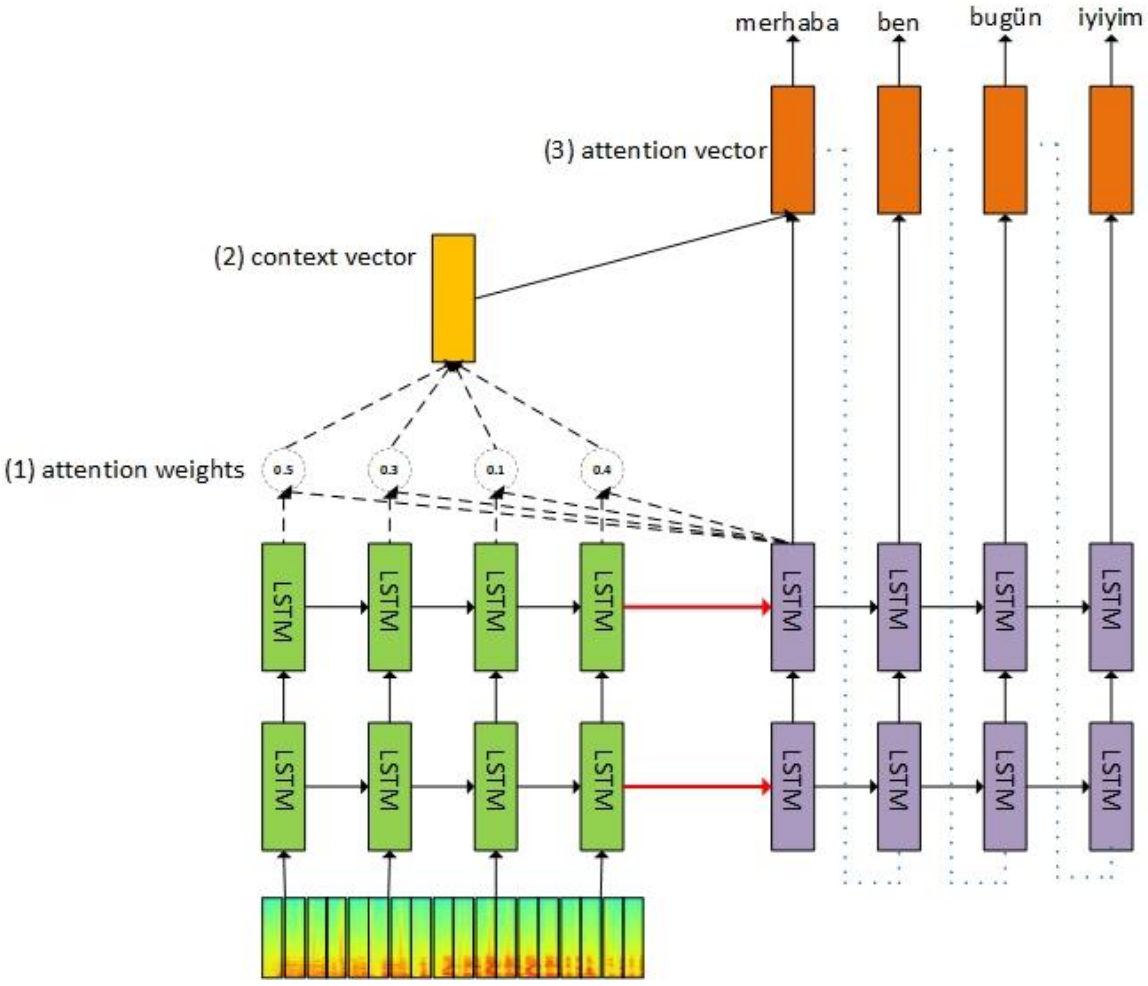
$$\alpha_{ts} = \frac{\exp(score(h_t, \hat{h}_s))}{\sum_{\acute{s}=1}^{S} \exp(score(h_t, \hat{h}_{\acute{s}}))} \tag{4.12}$$

- Odaklanma ağırlıklarını kullanarak bağlam vektörünü (context vector) hesaplıyoruz (Şekil 4.15 adım 2):

$$c_t = \sum_s \alpha_{ts} \hat{\mathrm{h}}_s \tag{4.13}$$

- Bağlam vektörü ile çıktıların saklı durumu birleştirilip odaklanma vektörü elde ediliyor (Şekil 4.15 adım 3):

$$a_t = f(c_t, h_t) = \tanh(W_c[c_t; h_t]) \tag{4.14}$$



Şekil 4.15 Ses sinyalini kelimelere çeviren ve odaklanma mekanizmasının adımlarını gösteren RNN yapısı.

score fonksiyonu çıktıdaki saklı durum ile girdideki saklı durum arasındaki karşılaştırma için kullanılıyor. Bu karşılaştırma farklı fonksiyonlar ile gerçekleştirilebilir ve bunlardan en önemlisi aşağıdaki gibidir:

$$score(h_t, \hat{\mathrm{h}}_s) = \begin{cases} h_t^T W \hat{\mathrm{h}}_s \\ v_a^T \tanh(W_1 h_t + W_2 \hat{\mathrm{h}}_s) \end{cases} \quad (4.15)$$

4.9. Bağlantıcı zamansal sınıflandırıcı

Konuşma tanımadaki akustik model eğitiminde, ses sinyaliyle yazısı arasındaki zamansal ilişkinin bulunması gerekmektedir. El yordamı ile de eğitim setindeki konuşmalar ve yazıları arasında bu şekilde bir hizalamanın oluşturulması zaman maliyeti açısından olası değildir. Diğer taraftan da, bu hizalamayı bulmadan sesi yazıya çevirebilen bir modelin elde edilmesi mümkün değildir. Dolayısıyla, bir ses için olası olan tüm karakter hizalamalarını göz önünde bulundurarak en optimal olanını bulan bir algoritmaya ihtiyaç duyulmaktadır.

İlk denemede bu şekilde bir yöntem önerilebilir; sesteki her on çerçeveyi bir karakter ile eşleştirelim. İnsanların konuşma tarzı/hızı farklı olduğu için çerçeveler ile karakterler arasındaki ilişki hep aynı olmayıp bu yöntem ilk deneylerde başarısız olacaktır. Başka bir yöntem olarak ses ile yazı arasındaki hizalamanın el ile yapılmasıdır. Modelleme açısından bu yöntem iyi çalışabilir ama büyük bir veri seti üzerinde tüm sesleri hizalamak oldukça zaman alıcı bir süreç olup mantıklı bir yöntem değildir. Bu sorun sadece konuşma tanıma için çözülmesi gerekmiyor ve el yazısını tanıyan bir model eğitimi için de yazı ile karakterler arasında hizalama yapılması gerekmektedir (Şekil 4.16). Bağlantıcı zamansal sınıflandırıcı (connectionist temporal classification - CTC) algoritması dizi halindeki girdi ve çıktı arasındaki hizalamayı yapmak için kullanılıp temel olarak HMM modeline benzer bir yapısı vardır.

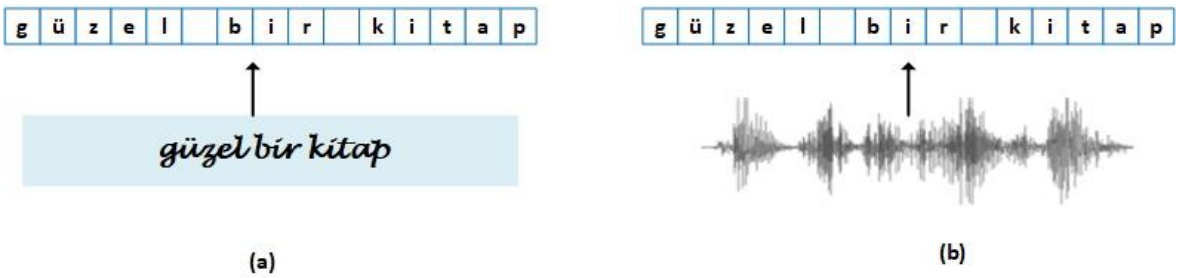
CTC tekniği dinamik uzunluktaki **x** dizisini dinamik uzunluğu olan **y** dizisine eşleştirmek için kullanılan bir algoritmadır [113]. Çıktıdaki **y** dizisinin uzunluğu |y| girdiden daha kısa olan durumlarda bu algoritmayı hizalamak için kullanabiliriz. **a** ile gösterilen hizalamayı yapabilmek için boşluğu gösteren "-" sembolü kullanılıyor. Matematiksel bir gösteriş ile:

$$P(a|x) = RNN(x) \tag{4.16}$$

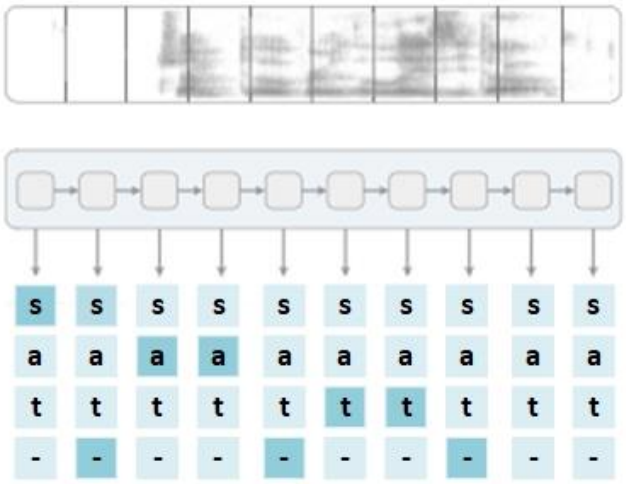
Bu formülde **x** dizisi ile **a** dizisi aynı uzunluğa sahip olup RNN fonksiyonu farklı yöntemler ile gerçekleştirilebilir [103]. Tüm çıktılar üzerinde toplayıp normalize edilirse:

$$P(y|x) = \sum_{a \in \beta^{-1}(y)} p(a|x) \tag{4.17}$$

β fonksiyonu çıktıdaki boşlukları atıp yeni bir dizi ile tekrarlıyor. Örneğin, β(*a-ab-*) = β(*-aa--abb*) ve her bir **a** farklı bir hizalamayı gösteriyor. P(y|x) olasılığı, dinamik programlamayı kullanarak tüm hizalamaları göz önünde bulundurup optimize edilebilir (Şekil 4.17) [113].



Şekil 4.16 El yazısını tanıyan bir model eğitmek için yazı ile karakterler arasındaki eşleşme model tarafından öğrenilmesi gerekiyor (a). Aynı problem konuşma tanıma için de geçerli olup ses ile yazı arasındaki hizalamanın bulunması gerekmektedir (b).



Şekil 4.17 saat kelimesi için olası hizalamalar CTC tarafından göz önünde bulundurulup her bir hizalama için skor hesaplanıyor.

CTC fonksiyonu ve bunun farklı versiyonları birçok konuşma tanıma sistemlerinde uygulanmış olup başarılı sonuçlar da elde edilmiştir [12, 70, 71, 114, 115]. Ancak, tüm bu çalışmalar güçlü bir dil modeli kullanarak tanıma sonuçlarını iyileştirip akustik model tek başına dil modelini öğrenemiyor. Bunun nedeni de, CTC fonksiyonu HMM gibi girdiler arasında bağımsızlığı varsayıp uzun bağımlılıkları modelleyememesidir [111].

5. UÇTAN-UCA KONUŞMA TANIMA

Bu bölümde, uçtan uca (end-to-end) bir konuşma tanıma modelinin yapısı ve deney sonuçları verilecektir. Kullanılan model, diziden diziye (sequence-to-sequence – seq2seq) bir sinir ağıdır ve ses girdisini direkt yazı çıktısına çevirebilen bir yapıya sahiptir. Bu yapı, klasik HMM tabanlı modellere göre çok daha basittir ve minimum müdahale ile ve hızlı bir şekilde çalışabilen bir sistem elde edilebilir. Ayrıca, akustik, dil ve okunuş modelleri hepsi birleştirilip tek bir model yapısında eğitildiği için dilin farklı özellikleri göz önünde bulundurularak bir model eğitimi yapılacak. Önerilen bu model, Dinle, Odaklan ve Yaz (DOY) olarak üç farklı bileşenin birleştirilmesi sonucunda elde ediliyor.

5.1. Uçtan Uca Konuşma Tanıma Modeli

DOY, uçtan uca eğitilen bir konuşma tanıma modelidir. Bu model sesi direkt karakterlere çevirebilen diziden diziye sinir ağı tipini kullanıyor [106]. Bu sistemde, ayrı bir dil modeli ve okunuş modeli kullanılmayıp bu bilgiler DOY modeli içerisinde sesin akustik özellikleri ile birlikte öğrenilmektedir. Ayrıca bu model, çıktıdaki karakterler ile girdideki ses girdisi arasında herhangi bir bağımsızlık varsayımında bulunmamaktadır. Bu model, odaklanma mekanizması (attention mechanism) bulunan ve girdi dizisini çıktı dizisine çevirebilen derin öğrenme yöntemlerine dayanmaktadır [93, 106, 116, 117].

DOY modeli içerisinde, kodlayıcı bir RNN yapısı dinleme yapıp çözücü olan bir diğer RNN de duyulan sesleri yazıya döküyor. Dinleme tarafındaki RNN, girdideki sesi işleyip üst düzeydeki öznitelik vektörlerini oluşturduktan sonra, çözücü tarafındaki RNN bu öznitelik vektörlerini kullanarak önceki çıktıları ve ses girdilerini de göz önünde bulundurup karakter dizisini üretiyor. Her adımda, RNN kendi içerisindeki durum bilgisini kullanarak odaklanma mekanizmasını yönlendirip üst düzey özniteliklerden bir bağlam vektörü (context vector) elde etmeye çalışıyor [116, 117]. Bu vektörü hem kendi iç durumunu (internal state) güncellemek için hem de doğru karakterleri üretmek için kullanıyor. Tüm model birlikte (jointly) ve doğru karakterleri üretmesi için zincir türevleri kullanarak eğitiliyor. Ayrıca bu model, konuşma tanıma sistemlerinde bulunan diğer bileşenleri de kendi içerisinde barındırıp tamamen uçtan uca eğitilmektedir.

5.2. Model

Bu bölümde DOY'un matematiksel tanımı verilecektir. Girdi vektörü olan x = (x_1, …, x_T) sesteki filter bank'ları gösterip y = (<sos>, y_1, … , y_s , <eos>) de çıktıdaki karakterleri gösteriyor. Karakter listesi, a'dan z'ye Türkçe karakterleri, sayıları ve noktalama işaretlerini içeriyor. Ayrıca, <*sos*> ve <*eos*> etiketleri de dizi başlangıç ve bitişini gösteriyorlar.

DOY modeli, çıktıdaki y karakterini önceki çıktılar ve sesteki öznitelik vektörünü göz önünde bulundurarak olasılıktaki zincir kuralı ile modelliyor:

$$P(\mathrm{y}|\mathrm{x}) = \prod_i p(y_i|\mathrm{x}, y_{<i}) \tag{5.1}$$

Bu objektif fonksiyonu, ayrımcı (discriminative) ve uçtan uca eğitilen bir modeli ifade ediyor.

DOY iki alt modülden oluşuyor: Dinleyici ve yazıcı. Dinleyici, kodlayıcı bir RNN yapısı olup ses girdisi üzerinde dinleme yapıyor. Dinleme süreci, sesi üst düzey bilgileri içeren **h** = (h_1, …, h_U) U≤T vektörüne dönüştürmektedir. Yazıcı taraf da odaklanma mekanizmasını kullanarak bu vektörleri çözüp karakter listesine dönüştürüyor (Şekil 5.1):

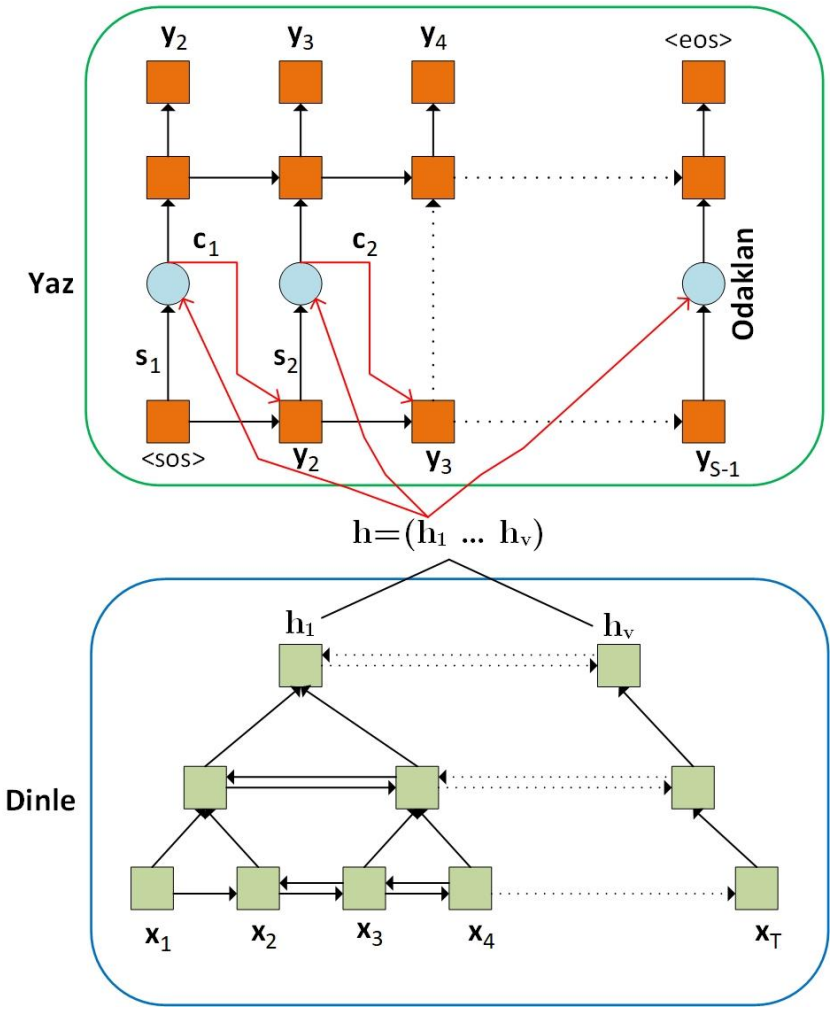
$$h = \mathrm{Dinle}(\mathrm{x}) \tag{5.2}$$

$$P(y_i|\mathrm{x}, y_{<i}) = \mathrm{OdaklanYaz}(y_{<i}, h) \tag{5.3}$$

5.2.1. Dinleme

Dinleme modülü, çift taraflı LSTM'leri (bidirectional long short term memory - BILSTM) kullanarak dinleme aşamasını gerçekleştiriyor. Bu yapı piramit şeklinde kurulup girdideki yüzlerce veya binlerce **T** ses çerçevesini **U** uzunluğundaki **h** vektörüne dönüştürüyor. BILSTM'leri direkt olarak kullanıp piramit yapısını kullanmamak eğitim süresini çok uzatıp 400 saatlik bir derlemden model eğitmek aylarca sürebiliyor. Bunun nedeni de binlerce ses çerçevesinin model tarafından işlenip gerekli bilgileri çıkarması gerektiğinden kaynaklanıyor. Bu sorunu aşmak için

üst üste yığılan BILSTM katmanlarında, her katmanda kullanılan ünite sayısı yarıya düşürülüyor.



Şekil 5.1 Odaklanma mekanizmasını kullanan dinleme ve yazma modülleri.

Normal bir BILSTM yapısı bu şekilde bir hesaplama yapıyor:

$$h_i^j = \mathrm{BLSTM}(h_{i-1}^j, h_i^{j-1}) \tag{5.4}$$

Piramit yapısındaki BILSTM katmanları da aşağıdaki gibi bir hesaplama yapıyor:

$$h_i^j = \mathrm{pBLSTM}(h_{i-1}^j, [h_{2i}^{j-1}, h_{2i+1}^{j-1}]) \tag{5.5}$$

İlk BILSTM katmanı üzerine üç BILSTM katmanı daha eklenerek girdideki çözünürlük düşürülmüştür. Bu da sistemin hem eğitim hem de test aşamasındaki süresini yaklaşık 8 kat hızlanılıyor. Ayrıca, odaklanma modülü daha az zaman damgası üzerinde işlem yaptığı için girdideki bağımlılıkları daha iyi bir şekilde modelleyebiliyor. Buna benzer yapılar literatürde hiyerarşik RNN'ler [118] ve katlamalı sinir ağları (convolution neural network - CNN) olarak bulunmaktadır.

5.2.2. Odaklan ve Yaz

Odaklan ve yaz modülü odaklanma mekanizmasını kullanan RNN-LSTM yapısını kullanmaktadır [117]. Yazıya dökmedeki her adımda o ana kadarki çıktılar ve girdi sinyali göz önünde bulundurularak yeni bir karakter çıktısı çözücü tarafından üretiliyor. y_i çıktısı çözücünün durumu olan s_i ve bağlamı gösteren c_i değerlerine bağlı olarak elde ediliyor. Ayrıca, çözücüdeki s_i durumu bir önceki durum s_{i-1} , bir önceki karakter çıktısı y_{i-1} ve önceki bağlam c_{i-1} değerlerine bağlı olarak değişiyor. Bağlam bilgisini içeren c_i odaklanma mekanizması tarafından oluşuyor:

$$c_i = \text{AttentionContext}(s_i, \text{h}) \tag{5.6}$$

$$s_i = \text{RNN}(s_{i-1}, y_{i-1}, c_{i-1}) \tag{5.7}$$

$$P(y_i|\text{x}, y_{<i}) = \text{CharacterDistribution}(s_i, c_i) \tag{5.8}$$

CharacterDistribution fonksiyonu son katmanında *softmax* uygulanan çok katmanlı bir sinir ağını temsil edip RNN ise iki katmandan oluşan bir LSTM'dir.

Her bir *i* zaman diliminde odaklanma mekanizması olan AttentionContext fonksiyonu bir bağlam vektörü olan c_i üretiyor. Bu vektör, girdi ile birlikte çıktı olarak üretilmesi gereken karakteri belirliyor. Odaklanma mekanizması içerik tabanlıdır; çözücünün durumunu gösteren s_i ile üst düzey öznitelik olan h_u değerlerini birleştirip odaklanma vektörü olan α_i üretiyor. Daha sonra h_u vektörleri α_i ile lineer olarak harmanlanıp c_i üretilmektedir.

5.2.3. Optimizasyon

Çıktıdaki karakterlerin logaritmik olasılığı artırılacak şekilde model parametreleri eğitiliyor:

$$\theta = max_\theta \sum_i \log P(y_i|\mathrm{x}, y_{<i}; \theta) \tag{5.9}$$

Ancak, modelin eğitim ve deney aşaması arasında bir tutarsızlık var. Modelin test zamanında doğru çıktıları bilmediğimiz için çözücü taraf sürekli ürettiği bir önceki çıktıyı tüketmektedir. Dolayısıyla, eğitim aşamasında belli bir oranda modelin ürettiği çıktılar da girdi olarak kullanılıp hatalı çıktılara karşı model dayanıklı hale getirilmektedir. Ayrıca, seq2seq modellerin aşırı uyma problemine de çok müsait bir yapıları var. Örneğin, Türkçe verisi ile yaptığımız ilk deneyde eğitim setindeki hata oranımız sıfıra yakın oldu. Fakat test setinden seçilen cümleler için oldukça yüksek bir hata oranı elde edildi. Bu sorunu engellemek için, eğitim aşamasında da modelin kendi çıktılarını kullanan zamanlanmış örnekleme (scheduled sampling) yöntemi kullanıldı. Bu yöntem, eğitilen modeli yaptığı hatalara dayanıklı bir hale getirip bu tür hataları göz ardı edebiliyor:

$$\tilde{\theta} = max_\theta \sum_i \log P(y_i|\mathrm{x}, \tilde{y}_{<i}; \theta) \tag{5.10}$$

5.2.4. Deşifre ve Skorlama

Deşifre zamanındaki amacımız **x** sinyaline ait en benzer karakter listesini bulmaktır:

$$\hat{y} = argmax_y \log p(\mathrm{y}|\mathrm{x}) \tag{5.11}$$

Bunu gerçekleştirmek için basit bir soldan-sağa alan araması (beam search) yapılmıştır [106].

Bunun yanı sıra, bir dil modeli de ara sonuçları skorlamak için kullanılıp sonuçlar biraz daha iyileştirilmiştir. Denemelerimiz sonucunda, modelin kısa cümlelere karşı bir eğilimi olduğunu fark ettik. Dolayısıyla, deşifreyi gerçekleştiren denklemi çıktının uzunluğuna göre normalize edip daha sonra dil modeli ile birleştiriyoruz:

$$s(y|x) = \frac{\log p(y|x)}{|y|_c} + \lambda \log p_{LM}(y) \quad (5.12)$$

λ katsayısı dil modelinin deşifre zamanındaki ağırlığı gösteriyor. Bu değer doğrulama seti (validation set) üzerinde yaptığımız deneylerle belirlenmektedir.

5.3. Deneyler

Deneyler için veri hazırlama bölümünde özellikleri verilen yaklaşık 100 saatlik bir konuşma derlemi kullanılmıştır. Veriler üzerinde sentezleme [119] yöntemleri uygulayarak eğitim seti için 382 saatlik bir veri seti hazırlandı. Ayrıca doğrulama ve test seti olarak da birer 7 saatlik ses verisi ayrıldı. Öznitelik vektörü olarak her 10ms'lık pencerelerde log-mel filterbank katsayıları hesaplanıp akustik bilgiler olarak model eğitiminde kullanılmıştır.

Konuşmalardaki metinler normalize edilip kelimelerdeki karakterler küçük harfe çevrilmiştir. Noktalama işaretlerinden: Nokta, boşluk, soru işareti, kesme işareti ve virgül karakterleri tutulup tanınmayan sözcükleri ise <*unk*> ile etiketlendi. Daha önce de belirtildiği gibi cümlelerin başlangıç ve bitişini belirleyen <*sos*> ve <*eos*> etiketleri cümlelerde kullanıldı.

Eğitim setinin 86 saatlik bir kısmı Türkçe radyo haberlerinden elde edilmiştir [2]. Bu veri ile HMM/GMM tabanlı bir modelden elde edilen en iyi sonuçta kelime hata oranı %21.3 olarak raporlanmıştır. İlgili çalışmadaki sonuç baz alınıp, aynı eğitim seti ve diğer verileri kullanarak da model eğitimi sonuçları bu tezde raporlanmıştır.

Dinleme tarafında, girdi vektörünü işleyen üç katmanlı ve 512 boyutlu BILSTM ağı yerleştirilmiştir. Bu işlem, veri çözünürlüğünü $2^3 = 8$ kat sayısı ile düşürüp eğitimlerin hem hızlanmasına hem de aşırı uymanın engellenmesine yol açıyor. Model parametreleri rastgele ve Uniform dağılımı olan U(-0.1, 0.1) ile başlatılmıştır.

Tensorflow [120] çerçevesindeki (framework) Asynchronous Stochastic Gradient Descent (ASGD) algoritmasını kullanarak model eğitimleri yapılmıştır. Öğrenme oranı (learning rate) 0.2 seçilip geometrik azaltma değeri (momentum) 0.98 oranı ile her 1 M cümlede uygulanmıştır. Eğitimlerin daha da hızlanması için ses dosyaları uzunluklarına göre kümelenip eğitime dahil edilmiştir [106]. Eğitimlerin hızlı

sonuçlanması için 4 adet GTX 980 Ti GPU kartı kullanılmıştır ve yaklaşık dört günde bir model eğitimi tamamlanmaktadır.

TBN [2] çalışmasındaki veri ile eğitilen MMI [121] modelinin kelime hata oran %21.3 olarak raporlanmıştır. Aynı veri ile eğitilen DOY modelinin hata oranı %22.1 olup dil modeli skorlamasından sonra %20.8'e düşmüştür (Çizelge 5.1). Dil modeli olarak [2] çalışmasındaki kelime tabanlı ve 184 M kelimeden oluşan bir metin derleminden *3-gram* eğitilmiştir. Bu sonuçlar, TBN çalışmasındaki eğitim setinin küçük olup (86 saat) uçtan uca bir model eğitimi için yetersiz kaldığını göstermektedir. Dolayısıyla, veri sentezleme yöntemi ile eğitim verisi hacmi artırılıp model eğitimleri daha büyük veri üzerinde tekrarlanmıştır (Çizelge 5.1).

Çizelge 5.1 Üç farklı eğitim seti ile yapılan model karşılaştırmaları. Baz model, [2] çalışmasında önerilen modeldir. Dinle, Odaklan, Yaz (DOY) modeli uçtan uca eğitilip dil modeli skorlaması ve sampling yöntemleri ile denenmiştir.

Train Set	**Model WER**				
	Baseline	DOY	DOY+LM	DOY+Sampling	DOY+Sampling+LM
TBN **[2]**	%21.3	%22.1	%20.8	%21.8	%20.8
TBN_Aug	%18.4	%16.8	%15.7	%16.2	%15.2
All	%17.8	%16.0	%15.3	%15.6	%14.4

Daha önce 5.2.3 bölümünde de belirtildiği üzere bu yapıda eğitim aşaması ile deney aşaması arasında bir tutarsızlık var. Modeli eğitirken, çözücü taraf her adımda girdi olarak doğru karakter listesini alıp bir sonraki adıma geçiyor. Ancak, test aşamasında çözücü taraf bir önceki adımın çıktısını kendi girdisi olarak tüketiyor. Dolayısıyla, çıktılardaki hata zincir bir şekilde tüm adımlara yansıyıp model çıktıları sonuna kadar yanlış devam edebiliyor. Bu sorunun önüne geçmek için, çıktıların %15'lik kısmını kullanıp bir sonraki adımın girdisi olarak kullanıyoruz. Bu yöntem sistemi kendi hatalarına karşı dayanıklı hale getiriyor. Örnekleme yöntemi Çizelge 5.1'deki tüm veri ile yapılan DOY modelinin hata oranını %16.2'den %15.6'ya düşürüyor. Çizelge 5.1'deki tüm deneylerde *n-best* listesindeki ilk 32 liste üzerinde dil modeli skorlaması yapılmıştır.

Sonraki bölümlerde dil modeli skorlamasını etkileyecek olan alan araması (beam search) araştırılacaktır. *N-best* listesi üzerinde yapılan aramaların *n* ile ne kadar bağlantılı olup ve sistemin hata oranını ne kadar etkileyeceği sonucu da verilecektir.

5.3.1. Çerçeve Örneklenmesi

Girdi sinyali 10ms'lik çerçeveler bölündüğü zaman yüzlerce girdi vektörü üretiliyor. Bu vektörlerin hepsini girdi olarak seq2seq bir modelde kullanmak hem modelin eğitim aşamasını hem de deşifre aşamasına yavaşlatıyor. Ayrıca sinyale ait tüm çerçeveleri kullanmak, veriye özel gereğinden fazla bilginin model tarafından öğrenilmesi ve aşırı uyma problemine de yol açabiliyor [122, 123]. Ses sinyalindeki çerçeveler üzerinde örnekleme yapıp her iki çerçeveden birisi atılmıştır. Bu yöntemle girdi vektörü sayısı yarıya indirilip deney sonuçları Çizelge 5.2'de raporlanmıştır.

Çizelge 5.2 Çerçeve örnekleme ile elde edilen sonuçlar.

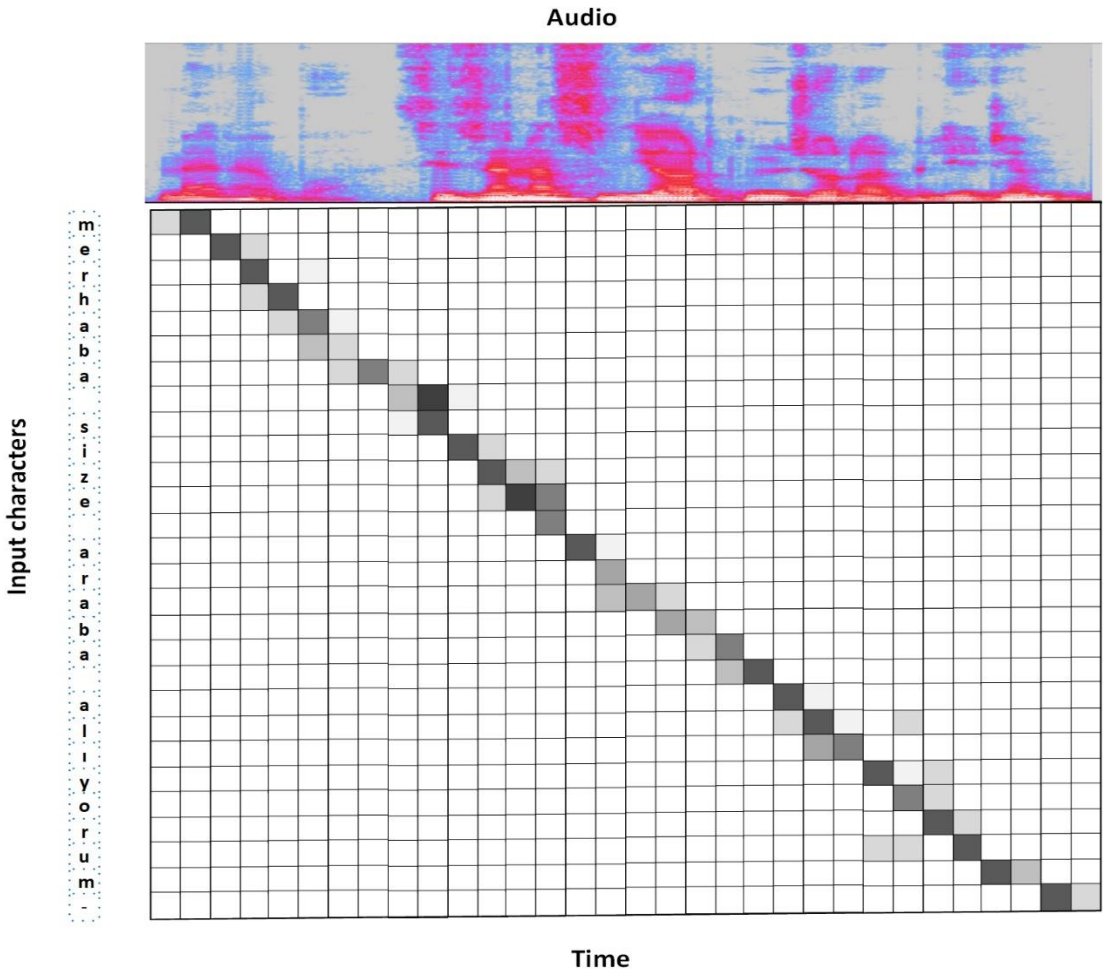
Train Set	**Frame Subsampling Model WER**				
	Baseline	DOY	DOY+LM	DOY+Sampling	DOY+Sampling+LM
TBN **[2]**	%21.3	%21.9	%20.5	%21.7	%20.5
TBN_Aug	%18.4	%16.7	%15.3	%16.12	%15.0
All	%17.8	%15.8	%15.1	%15.3	%14.2

Tüm veri ile eğitilen ve dil modeli kullanarak skorlama yapılan deneyde %1.38 relatif iyileşme sağlanarak hata oranı Çizelge 5.1'deki %14.4'ten %14.2'ye düşmüştür.

5.3.2. Odaklanma Görselleştirmesi

Bağlam tabanlı odaklanma mekanizması karakterler ile ses arasında belirgin bir hizalama yapabiliyor. Odaklanmanın ses sinyali üzerindeki dağılımını her karakter çıktısı üzerinde görselleştirirsek bu ilişkiyi görebiliriz. Şekil 5.2'de "merhaba size araba alıyorum" cümlesi ile ses sinyali arasındaki hizalama odaklanma mekanizması ile üretilmiştir. Odaklanma mekanizması her karakter için ses üzerindeki odaklanması gereken aralıkları doğru tespit edip karakterlerle çerçeveler arasındaki hizalamayı da göstermektedir. Ayrıca, cümlelerin başı ve sonu da model tarafından doğru tespit edilebiliyor. Odaklanma mekanizması *merhaba* ile *araba*

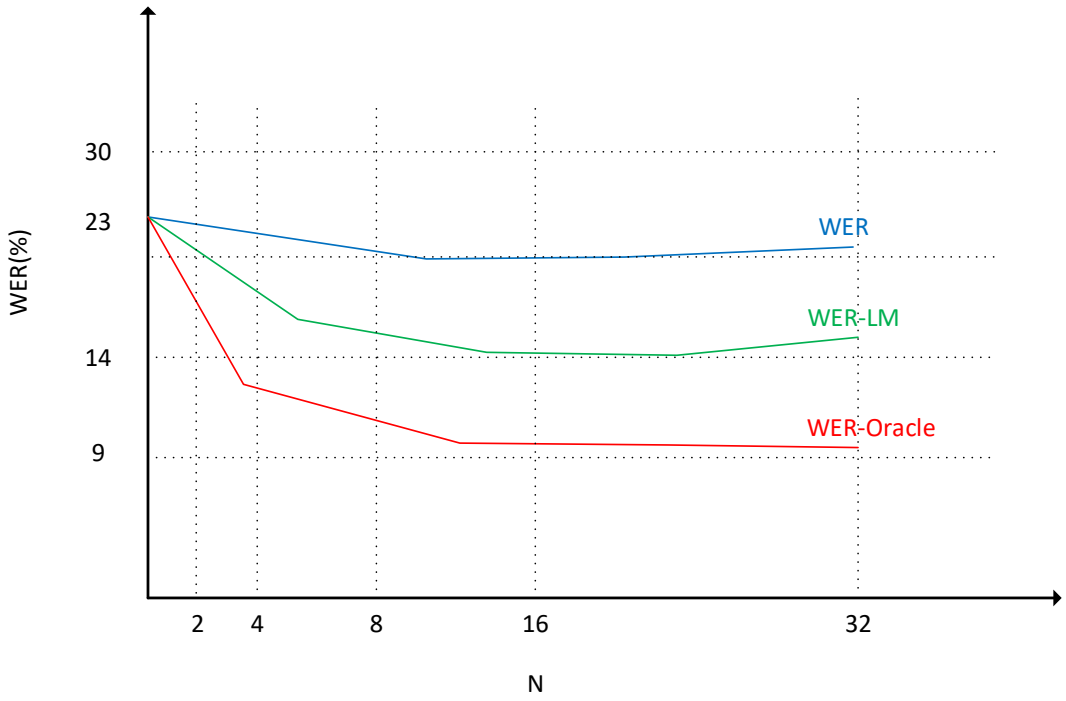
kelimeleri arasındaki akustik benzerlikten dolayı bu kelimelerin dağılımı biraz daha açık renkle çıkmıştır.



Şekil 5.2. DOY modeli ile üretilen karakter çıktısı ve ses sinyali arasındaki hizalama. Bu görselde, “merhaba size araba alıyorum” cümlesi v ses sinyali arasındaki hizalama görülmektedir.

5.3.3. Alan Araması Genişliği

Bu kısımda, alan araması (beam search) genişliğinin model başarısı üzerindeki etkisi araştırılmıştır. Deneyler hem dil modeli hem dil modeli kullanmadan yapılıp *n-best* listesindeki n değerinin büyüklüğü incelenmiştir.



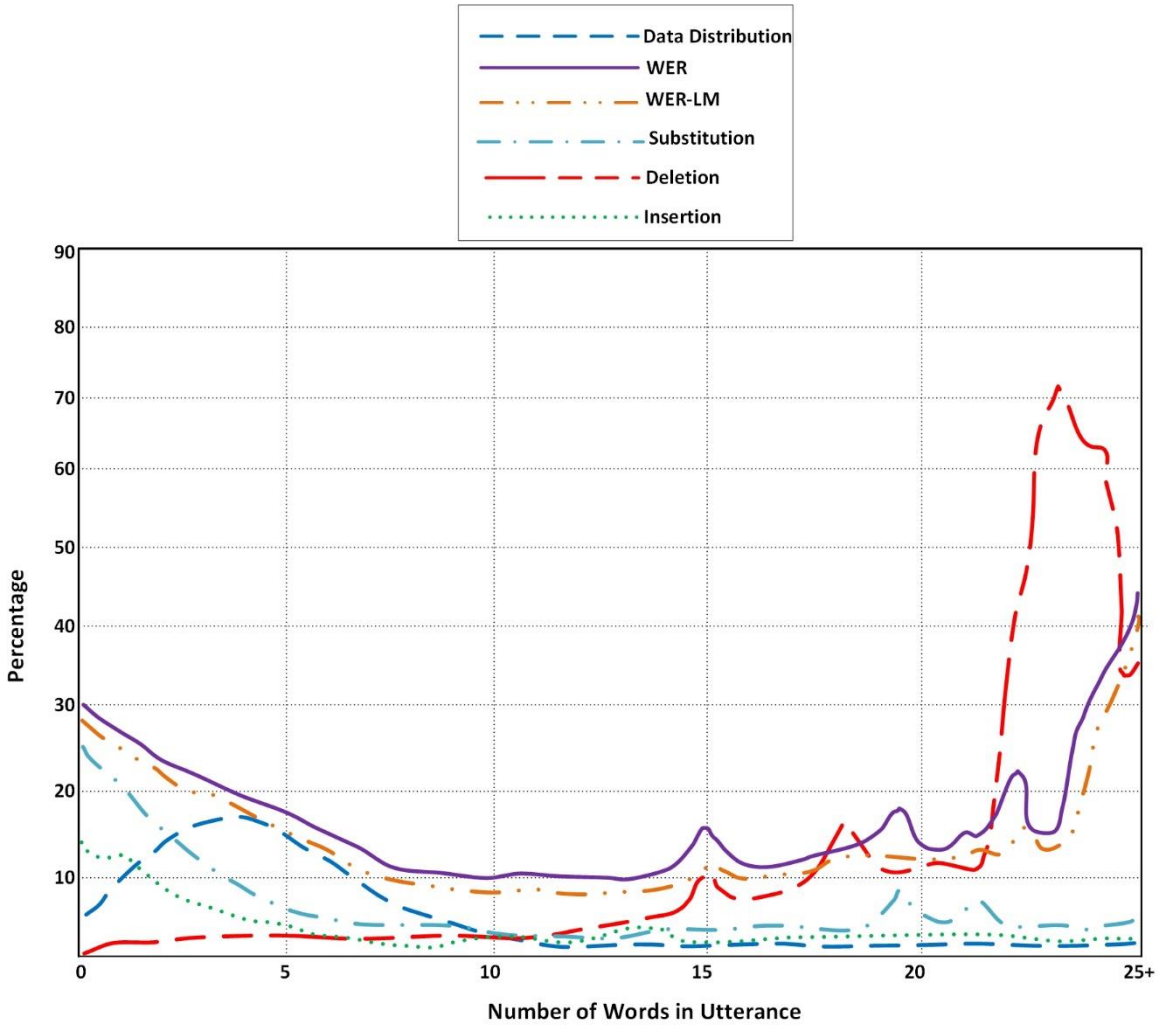
Şekil 5.3 n-best listesindeki n değerinin hata oranına olan etkisi. Model eğitimi tüm veri ile yapılıp büyük test set kullanılmıştır.

Şekil 5.3'te, *n* değerinin kelime hata oranına olan etkisi çizilmiştir. Hata oranı n=16'ya kadar kararlı bir biçimde düşüp 16 ile 32 arasında kayada değer bir fark gözlemlenmemiştir. Kelime hata oranı tüm veri ile eğitilen DOY+Sampling modelinde n=32 olduğu zaman %15.6 olup dil modeli skorlamasından sonra %14.4'e düşmüştür.

5.3.4. Cümlelerin Uzunluk Etkisi

Modelin doğruluk oranı ile cümlelerdeki kelime sayısı arasındaki ilişkiyi anlamak için cümledeki kelime sayısına göre bir test yapıldı. Eğitim setindeki örnekler daha çok kısa cümlelerden (3-5 kelime arası) oluştuğu için uzun cümlelerdeki başarının biraz daha düşük olmasını bekliyoruz. Hata türü ile cümledeki kelime sayısı arasındaki ilişki Şekil 5.4'te verilmiştir.

Uzun cümlelerdeki hata oranı daha çok kelimelerin silinmesinden (deletion error) kaynaklanıp sesin sonuna doğru kelimeler unutulmaya başlıyor. Ayrıca çok kısa cümlelerde (iki veya daha az kelime içeren) de hata oranı yüksek olup hataların büyük bir kısmı değiştirme ve eklemeden (substitutions and insertions) kaynaklanıyor.

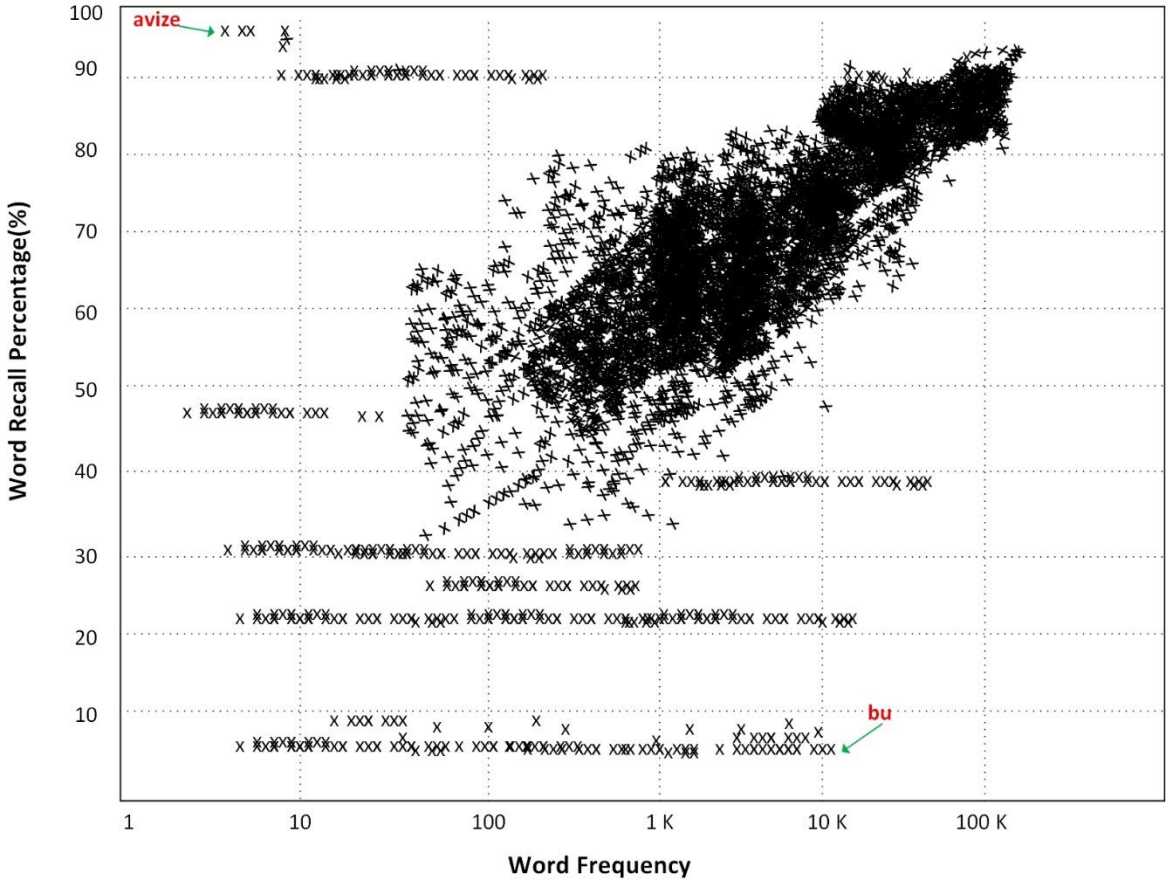


Şekil 5.4 Hatalar (ekleme, silme, değiştirme) ile cümledeki kelime sayısı arasındaki ilişki. Hata oranı herhangi bir sözlük veya dil modeli kullanmadan raporlanmıştır. Modelin hata oranı kısa cümleler ve uzun cümlelerde daha yüksek görülmektedir.

5.3.5. Kelime Sıklığı

Bu kısımda seyrek geçen kelimelerde modelin doğruluk oranı incelenmiştir. Metrik olarak Recall değeri kullanılıp bu değerin yüksek olması hatanın düşük olacağı anlamına gelecektir (kelimenin pozisyonundan bağımsız olarak). Şekil 5.5'te test setinde bulunan kelimelerin Recall değeri ile eğitim setindeki kelimelerin sıklığı arasındaki ilişki verilmiştir. Görüldüğü üzere, seyrek geçen kelimelerin standart sapması daha fazla olup Recall değeri daha düşüktür. Çok sık geçen "bu" kelimesi 20 bin kere eğitim setinde geçmiştir, ancak, Recall değeri %85 olarak kaydedilmiştir.

“bu” kelimesi çoğu yerde “şu” olarak yanlış yazılmıştır ve “şu” kelimesinin Recall değeri %95 olarak kaydedilmiştir. Dolayısıyla, dil modeli tarafında da bazı iyileştirmelerin yapılması gerekmektedir. Diğer taraftan da, sadece bir kere geçen “avize” kelimesinin Recall değeri %98 çıkmıştır. Demek ki, bir kelimenin doğru tanınması sadece veri setindeki sıklığı ile ilişkili olmayıp kelimenin akustik olarak belirleyici olması da önemlidir.



Şekil 5.5 Eğitim setindeki kelimelerin sıklığı ile test setindeki kelimelerin Recall metriği arasındaki ilişki.

5.3.6. Çıktıların Analizi

DOY modelinin üretebileceği çıktıları daha net analiz edebilmek için, bu bölümde birkaç örnek cümlenin analizi yapılmaktadır. Tüm deneylerde herhangi bir sözlük modeli veya dil modeli kullanılmamıştır.

Deneylerimizde, DOY modelinin aynı akustiğe sahip kelimeler için farklı yazılışları üretebileceğini görmekteyiz. Çizelge 5.3'te "merhaba üç nokta koy" cümlesine ait DOY modelinin ürettiği 14 farklı çıktı verilmiştir. Çıktılardaki sonuçlara göre, model, "üç nokta" ve "…" sözcüklerini ilk 3 tahmininde bulmuştur.

Çizelge 5.3 "merhaba üç nokta koy" cümlesinin farklı çıktıları.

Beam	Text	Probability (Log)
Truth	*merhaba üç nokta koy*	-
1	merhaba üç nokta koy	-0.5278
2	merhaba üç nokta köy	-0.8963
3	merhaba … koy	-0.8998
4	merhaba uç nokta koy	-1.3214
5	merhaba üç noktalama koy	-3.4589
6	araba üç nokta koy	-3.8978
7	merhaba araba nokta koy	-6.7821
8	araba üç nokta köy	-6.8589
9	araba … köy	-6.9974
10	araba uç noktalama koy	-7.1259
11	merhaba kuç nokta kuy	-7.4589
12	arabam ücu noktam köy	-7.5698
13	merhabam noktalam köyu	-8.5692
14	arabam … uç kuy	-8.6989

HMM tabanlı modellerde sadece sözlükte bulunan okunuşların sistem tarafından üretilmesi mümkündür. Ancak uçtan uca eğitilen DOY modeli eğitim setinde bulunmayan okunuşları da üretebiliyor. İnsan da daha önce hiç duymadığı bir kelimenin sesini duyduğu zaman o kelimenin yazılışına yakın kelimeyi yazabiliyor. Dolayısıyla, DOY modeli insanın konuşma tanıma yapısına daha yakın bir biçimde çalışabiliyor.

DOY modeli tekrarlanan kelimeleri de doğru bir şekilde tanıyabiliyor. Çizelge 5.4'te "bir iki altı sekiz sekiz sekiz" cümlesinin DOY tarafından ürettiği çıktılar bulunmaktadır. Modelde içerik tabanlı bir odaklanma mekanizması bulunduğu için tekrarlanan kelimelerin model tarafından daha az veya daha fazla yazılmasını bekliyorduk. Ancak, örnekten de anlaşıldığı gibi "sekiz" kelimesi cümle içerisinde üç defa tekrarlanmış olup DOY tarafından da üç kez tanınmıştır.

Çizelge 5.4 “sekiz” kelimesinin tekrarlandığı bir örnek.

Beam	Text	Probability (Log)
Truth	*bir iki altı sekiz sekiz sekiz*	-
1	bir iki altı sekiz sekiz sekiz	-0.4228
2	bir iki altı sekiz sakiz sekiz	-0.7865
3	bir iki alt sekiz sekiz sekiz	-0.7995
4	bir iki alt sakız sekiz sakiz	-1.2614
5	bir ikil altı sakız sekiz sekiz	-3.4781
6	biri iki altım sekiz sekiz sekiz	-3.9270
7	bir ikili at sekiz sekiz sakız	-6.7028
8	bir ikili atım sakiz sekiz sakız	-6.9524
9	birim ikili atım sekiz sekiz sakız	-6.9987
10	birim iki altı sakiz sakiz sakız	-7.5672
11	birisi ikili attı sakız sekiz sakız	-7.5898
12	birleş iki altım sekiz sekiz sakız	-7.6899
13	biri ikili atım sekiz sakla sakiz	-8.5601
14	bir ikili altmış saksı sekiz sakız	-8.8214

DOY modeli okunuş dilini yazı diline çevirmede de başarılı sonuçlar üretiyor. Türkçe’de çoğu zaman bazı harfleri yutarak kelimeleri söylüyoruz. Örneğin “merhaba” kelimesini “meraba” ve “nasılsınız” kelimesini “nasısınız” şeklinde telaffuz ediyoruz. “meraba nasısınız eve gidiyom” şeklinde okunan bir cümlenin model tarafından üretilen çıktı sonuçları Çizelge 5.6’da verilmiştir. Çıktılara baktığımız zaman model okunmayan harfleri de tahmin ederek “meraba”, “nasısınız” ve “gidiyom” kelimelerinin doğru yazılışlarını da üretip otomatik bir şekilde metin normalizasyonu gerçekleştirebiliyor. Eğitim setindeki metinler yazı dilinde ve doğru transkript edildikleri için model bir süre sonra okunmayan seslerin de doğru yazısını üretmeyi öğreniyor.

HMM tabanlı modellerde bu tür okunuşların sistem tarafından tanınabilmesi için Çizelge 5.5’te örnek verildiği gibi, okunuş sözlüğünde bir kelimenin çeşitli okunuşları bulunmalıdır. Ancak, bu okunuşların daha önceden ve konuşmadan bağımsız bir şekilde hazırlanması, dilin yapısını ve akustik özelliklerini göz önünde bulundurmadan hataya açık ve tamamen tahmine dayalı bir süreçtir.

Çizelge 5.5 HMM tabanlı sistemlerin sözlüğündeki farklı okunuşlar.

Word	Pron.1	Pron.2	Pron.3
merhaba	M E R H A B A	M E R A B A	M E R B A
gidiyorum	G İ D İ Y O R U M	G İ D İ Y O M	G İ D İ O M
nasılsınız	N A S I L S I N I Z	N A S I S I N I Z	N A S S I N I Z

Çizelge 5.6 “meraba nasısınız eve gidiyom” şeklinde okunan bir cümlenin çıktıları.

Beam	Text	Probability (Log)
Truth	*merhaba nasılsınız eve gidiyorum*	-
1	merhaba nasılsınız eve gidiyorum	-0.3258
2	merhaba nasılsınız eve gidiyoum	-0.5866
3	merhab nasısınız eve gidiyorum	-0.8225
4	merhaba nassınız eve gidiyorum	-1.5618
5	merba nasılsınız evi gidiyom	-3.6775
6	merhab nasıl eve gidiyorum	-3.8221
7	merhaba nasısınız evim gidyom	-6.6018
8	araba nasılsınız eve gidiyoruz	-6.9127
9	merhab ne nasılsınız evin gidiyorum	-6.9567
10	merba nasısın eve gidiyorum bu	-7.1235
11	merhaba o nasılsın evin gidiyorum	-7.5821
12	araba bu nasıl eve gidiyor	-7.6129
13	merhabalar nasılsınız evim gidiyor mu	-8.1647
14	arabalar nasıl eve gidiyor mu	-8.9215

5.4. Çıktıların Kısıtlanması

Öneki deneylerde modelin çıktıları karakter tabanlı olup Türkçe’deki harfler üretiliyordu. DOY modeli çıktı olarak karakterleri üretip daha sonra üretilen boşluklara göre bu karakterler birleştirildikten sonra kelimelere dönüştürülüyor. Çıktıdaki semboller sabit bir kelime listesi, fonem listesi veya karakter listesi veya bunların birleşiminden oluşabilir.

Kelime tabanlı bir model eğitmek girdilerin uzunluğunu kısaltıp modelin hem eğitim süresini hem de test süresinin kısa olmasını sağlıyor [124, 125]. Ancak kelime tabanlı modellerde, çıktı katmanındaki *softmax* fonksiyonunun uzunluğu Türkçe gibi sondan eklemeli diller için milyonlarca seviyesine ulaşıp kelimeler arasından seçim yapmayı zorlaştırıyor. Kelime sayısını kısmak da bazı kelimelerin modellenmemesine ve test zamanında sözlük dışı kelime (out-of-vocabulary - OOV) hatalarına sebep oluyor. Karakter tabanlı model eğitmek OOV sorununu ortadan

kaldırıp ancak girdilerin uzun olmasına neden oluyor. Bu da, hem eğitimin hem de test zamanının uzaması demektir. Ayrıca seq2seq modellerde, çok uzun girdiler modelin başarısını negatif yönde etkileyip modelin unutkanlığını da artırmaktadır [83, 116].

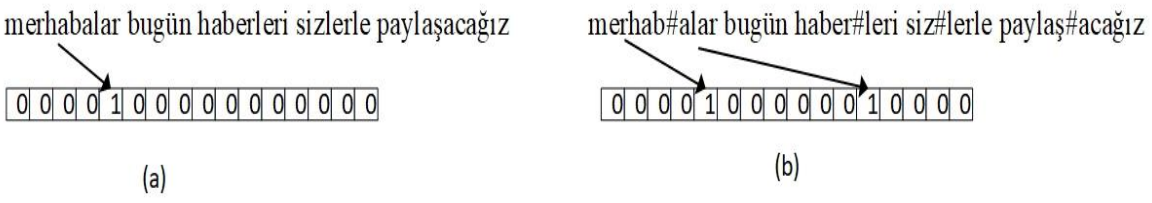
İngilizce'de 40 M kelimeden oluşan bir metin derleminde yaklaşık 200 K tekil kelime bulunuyor. Bu sayı, Fince ve Estonya dilleri için 1.8 M ve 1.5 M kelimeye ulaşıp Türkçe için aynı boyuttaki bir derlemde 735 K tekil kelime bulunuyor. Türkçe'de, kelimelere çekim ve yapım ekleri eklenerek yeni kelimeler üretilebiliyor. Örnek bir Türkçe cümle için kelime tabanlı ve kök-ek tabanlı bir gösterim aşağıdaki gibidir:

- Kelime tabanlı: derneklerinin öncülüğünde
- Kök-Ek tabanlı: dernek-lerinin öncü–lüğünde

5.4.1. Kök-Ek tabanlı Model Eğitimi

DOY modelindeki çıktıların uzunluğunu kısmak için ve OOV hatalarının azaltılması açısından kök-ek tabanlı bir model eğitimi denenmiştir. Eğitim setindeki konuşma derleminde 1.5 M kelime bulunup tekil kelime sayısı 250 K kelimedir. Bu da demektir ki, modele verilen one-hot 250 K'lık bir vektörün sadece tek bir elemanı 1 olup diğer elemanlar sıfırdır. Bu vektörün boyutu, 200 boyutlu bir gömme katmanından (embedding layer) geçirildikten sonra 200'e düşürülüyor [126]. Ancak, gömme katmanında 250 K × 200 boyutlu bir matrisin oluşturulması gerekiyor ve bu da ihtiyaç duyulan hafıza miktarı ve eğitim/test zamanını oldukça arttırıyor.

Kök-ek tabanlı model eğitimi için yazılardaki kelimeler [2] çalışmasındaki algoritmayı kullanarak kök-ek olarak ayrıştırılmıştır. Kelime tabanlı ve kök-ek tabanlı girdi vektörünün örmeği Şekil 5.6'da verilmiştir.



Şekil 5.6 (a) Kelime tabanlı girdi vektörü (b) kök-ek tabanlı girdi vektörü.

Bu Şeklin (a) kısmında daha önceki model eğitimlerinde kullandığımız kelime tabanlı girdi vektörü bulunuyor. “merhabalar” kelimesi tek başına alınıp one-hot vektöründeki tek bir eleman 1 yapılmıştır. Kök-ek tabanlı yöntemde ise hem kelimenin kökü için hem de ek kısmı için vektördeki elemanlar 1 yapılmıştır.

Kelimeleri kök-ek olarak ayrıştırdıktan sonra sözlük boyutu 150 K sözcüğe düşmüştür. Aynı yöntem dil modeli metni üzerinde de uygulanıp kök-ek tabanlı bir n-*gram* dil modeli ARPA formatında eğitilmiştir [127]. Bu yöntem ile eğitilen DOY modelinin sonuçları Çizelge 5.7’de verilmiştir. Model eğitimleri tüm veri seti üzerinde ve önceki bölümlerde anlatılan çerçeve örneklemesi ile yapılmıştır.

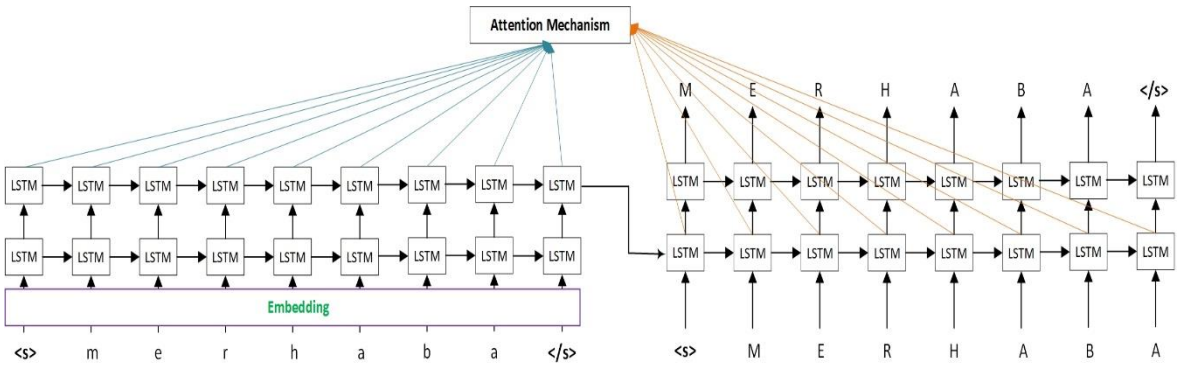
Çizelge 5.7 Kelime ve kök-ek tabanlı DOY modelinin sonuçları. Deneyler tüm konuşma verisi ile eğitilen DOY+FrameSubsampling yöntemi ile yapılmıştır.

Unit	Language Model Vocab.	n-gram	Decode Time	WER(%)
Word	50 K	3-gram	1.2	18.5
	76 K	3-gram	1.2	18.2
	200 K	3-gram	1.3	15.6
	300 K	3-gram	1.4	14.3
	500 K	3-gram	1.5	14.2
	500 K	4-gram	1.6	14.2
Stem-Postfix	46 K	3-gram	0.9	15.9
	200 K	3-gram	0.8	13.5
	200 K	4-gram	0.9	**13.5**

Kelime tabanlı deneylerde, dil modeli için kullanılan metin ve sözlük boyutu artırılarak *3-gram ve 4-gram* modelleri eğitilip DOY’daki deşifre sonuçları skorlanmıştır. Tekil kelime sayısı 50 K’lık deneyde kelime hata oranı %18.5’iken 500 K’lık sözlükteki hata oranı %14.2’ye düşmüştür. Hem sözlük dışı kelime sayındaki düşüş hem de dil modelinin daha büyük bir metinden eğitilmesi bu iyileşmeyi sağlamıştır. Kök-ek tabanlı modelde, kök ve eklerden oluşan sözlükteki ünite sayısı 46 K’dan 200 K’ya çıkarıldığı zaman kelime hata oranı %15.9’dan %13.5’e düşmüştür. Sonuçlardan da anlaşıldığı üzere hem kelime tabanlı modelde hem de kök-ek tabanlı modelde *n-gram* sayısını 3’den 4’te çıkarmak sonuçlar üzerinde bir etkisi yoktur. Tanıma süresi açısından ise kök-ek tabanlı model kelime tabanlı modele göre daha hızlı olup gerçek zamanlı (Real-Time) tanımadan 0.1 kadar hızlı tanıma yapabiliyor.

5.5. Kelime Okunuş Modeli

Bu kısımda Türkçe ve İngilizce kelimelerin okunuşunu üreten (phonetizer) bir seq2seq modelin yapısı ve eğitim süreci anlatılacaktır. Bu model yine de encoder-decoder yapısı ile çalışıp kodlayıcı taraf kelimedeki karakterleri girdi olarak alıp çözücü taraf da okunuşları üretmektedir. Türkçe fonetik bir dil olduğu için bu problem Türkçe için kolay bir problemdir. Ancak, sonuçların daha kolay bir şekilde incelenmesi ve daha sonra farklı diller için aynı yapının kullanılabilmesi için ilk denemeler Türkçe üzerinde yapılıp daha sonra İngilizce üzerinde de denemeler yapılmıştır.



Şekil 5.7 Kelimelerin okunuşunu üreten bir seq2seq model yapısı. Kodlayıcı taraf kelimedeki harfler one-hot türünden alıp kodladıktan sonra çözücü modeline iletmektedir.

Kodlayıcı taraf kelimelerdeki harfleri one-hot vektör türünden alıp gömme katmanından (Embedding Layer) geçirdikten sonra 50 boyutlu bir yoğun vektör (Dense Vector) olarak işliyor. Dizinin sonunu gösteren *</s>* etiketi kodlayıcıya verildikten sonra bu kısmın işi tamamlanıp bilgi vektörünü çözücüye iletiyor. Girdiler işlerken, odaklanma mekanizması ise kodlayıcı ve çözücünün ilk katmanları tarafından güncellenmektedir. Çözücü, *<s>* etiketini ilk girdisi olarak alıp çıktıları üretmektedir. Model eğitimi aşamasında, çözücü taraf kelimelerin doğru okunuşunu girdi olarak alıp doğru çıktıları üretmeye çalışıyor.

Model mimarisinde iki katmandan oluşan LSTM-RNN yapısı kullanılmıştır. Her iki tarafta sadece girdiyi soldan sağa işleyip tek yönlü bir RNN uygulamaktalar. LSTM'deki ünitesi sayısı 256 olup model parametreleri rastgele ve Uniform dağılımı

olan U(-0.1, 0.1) ile başlatılmıştır. Optimizasyon için Asynchronous Stochastic Gradient Descent (ASGD) algoritmasını kullanarak model eğitimi toplam 4 iterasyon olmak üzere yapılmıştır.

Türkçe dil modeli için kullandığımız metinin sözlüğü eğitim verisi olarak kullanılmıştır. Sözlük içerisinde 1 M tekil kelime bulunup rastgele olarak okunuşlar kontrol edildikten sonra hatalı kelimelerde düzeltmeler yapıldı. Okunuşların hazırlanması için basit bir kural kullanarak her harfin okunuşu harfin kendisi ve büyük karakter ile önüne yazılmıştır. Toplamda 2 bin yabancı kelimenin (çoğunluğu İngilizce) Türkçe okunuşu da el ile kontrol edilip önüne yazıldı (Çizelge 5.8).

İngilizce model eğitimi için CMU_DICT [8] sözlüğünde bulunan 133 bin kelime ve okunuşları eğitim verisi olarak kullanılmıştır. Türkçe'de kullanılan iki katmanlı RNN-LSTM ve odaklanma mekanizması bulunan bir model İngilizce verisi için de kullanıldı. Türkçe ve İngilizce sözlükteki kelimelerin %85'i eğitim için %5'lik bir kısmı doğrulama seti ve geriye kalan %5'i ise test için ayrılmıştır.

Çizelge 5.8 Okunuş modeli için hazırlanan eğitim verisindeki örnek kelimeler.

Word	Type	Pronunciation
BİLGİSAYAR	Türkçe	B İ L G İ S A Y A R
?	Türkçe Noktalama	S O R U
TBMM	Türkçe Kısaltma	T E B E M E M E
SERVER	İngilizce	S Ö R V I R
Infotech	İngilizce	İ N F O T E K

Model eğitimi 4 iterasyon sonunda tamamlanıp doğrulama setinde bir iyileştirme olmadığında eğitim sonlandırılmıştır. Test seti olarak ayrılan sözcükler üzerinde modelin ürettiği okunuşlar ile kelimelerin doğru okunuşu arasındaki fonem (phoneme error rate [128]) hata oranı ölçüldü. Türkçe fonetik bir dil olduğu için kelimelerin okunuşu yazıldığı gibidir. Dolayısıyla, fonem hata oranının Türkçe'de daha düşük olmasını bekliyoruz. Türkçe deneylerinde karşılaştığımız ilginç sonuçlardan birisi yabancı kelimeler için modelin doğru okunuşları üretebilmesidir. Örneğin, "international" kelimesi için "İ N T E R N E Y Ş I N I L" veya "TCD" gibi bir kısaltma için "T E C E D E" okunuşları üretildi. Eğitim setinde bulunan 2 bin İngilizce kelimenin okunuşundaki kuralları öğrenerek model yabancı kelimelerin Türkçe okunuşunu üretebiliyor.

Çizelge 5.9 Okunuş modelinin Türkçe ve İngilizce için olan PER (phoneme error rate) değeri.

Language	#Train Word	#Validation Word	#Test Word	#PER (%)
Turkish	1 M	50 K	50 K	%1.5
English	133 K	6 K	6 K	%5.6

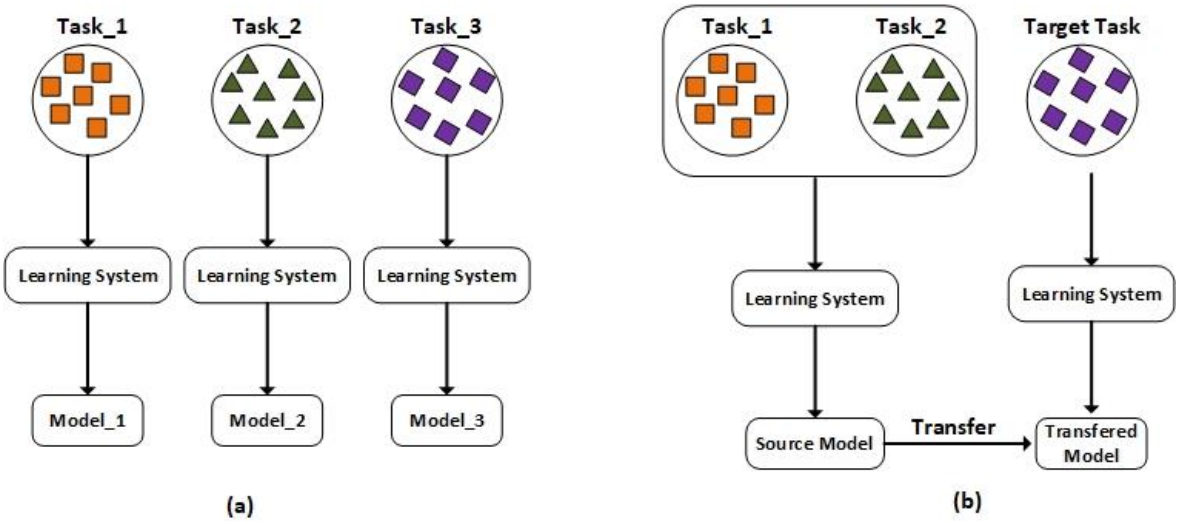
Bu model ev asistanları gibi cihazlarda sözlükte olmayan kelimelerin anlık olarak üretilmesinde kullanılabilir. Ayrıca HMM tabanlı sistemlerde sözlük kullanıldığı için kelimelerin okunuşu otomatik olarak bu model tarafından üretilebilir.

5.6. Transfer Öğrenme

Derin sinir ağlarındaki saklı katmanlar veriye ait bilgileri, farklı detay ve bilgi soyutlama seviyesinde tutmaktalar. İlk katmanlar daha detaylı ve problemden bağımsız bilgileri içerirken son katmanlara doğru bilgi seviyesi daha soyutlanmış olup probleme özel bilgiler bulunmaktadır. Dolayısıyla, benzer problemler için tasarlanan modellerde bulunan ilk katmanlar, problemler arasındaki ortak bilgileri içerdikleri için benzer yapıya sahip farklı bir görev için de kullanılabilirler [129].

Makine öğrenimi alanında, eğitim setinde bulunan verilerin ve öznitelik vektörlerinin aynı dağılımdan olması gerekiyor gibi bir algı var. Bu teori her zaman doğru olmayabilir ve farklı bir dağılımdan ve veri kümesinden eğitilen bir model başka bir veri setine adapte edilip iyi sonuçlar alınabilir [129]. Özellikle, bir göreve ait verinin az olması veya veri hazırlamanın pahalı olması durumlarda transfer öğrenme mantıklı bir yöntemdir. Ayrıca, hedef (target) ortamdaki doğruluk oranı düşükse bu ortamdan az miktarda veri toplayıp kaynak modeli (source model) bu veriye göre adapte etmek başarı oranını artırmaktadır.

Şekil 5.8'de makine öğreniminde kullanılan iki farklı yöntem gösterilmektedir. Birinci yöntemde farklı alanlar için farklı veri kümeleri oluşturup ayrı ayrı model eğitimleri yapılmaktadır. Bu yöntem, makine öğrenimi için klasik bir yöntem olup veri miktarı az olan bir görev için yeterince iyi bir modelin elde edilmesi mümkün değildir. Bu şeklin b kısmındaki yöntemde, veri hacmi fazla olan bir görev için model eğitilip daha sonra bu modeli başka amaca yönelik transfer edilmektedir.



Şekil 5.8 Farklı veri kümeleri için ayrı ayrı model eğitimleri (a). Başka veri kümelerinden eğitilen modeli kullanarak hedef göreve yönelik model transferi (b).

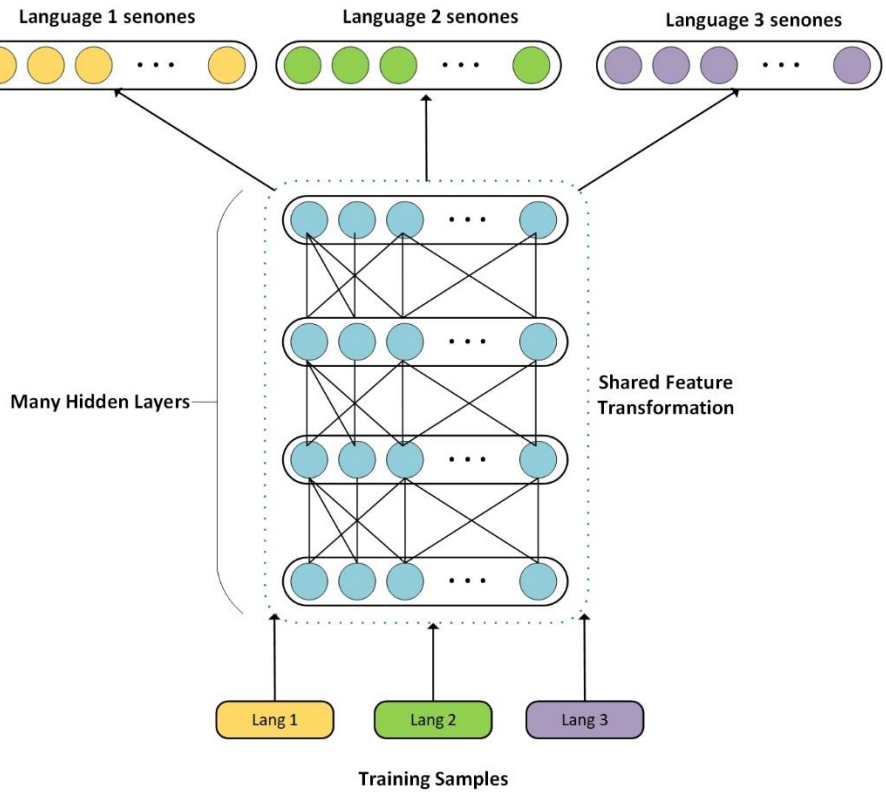
Konuşma tanıma alanında da, fonem ve ses benzerliklerini kullanarak farklı bir dildeki kaynak model yeni bir dile transfer edilebilir. Her ne kadar diller arasındaki kelimelerin okunuş ve sesleri bir birinden farklı olsa da, bu dillerde ortak öznitelik ve veri dağılımları bulunmaktadır [130]. Örneğin, fazla miktarda İngilizce verisi ile eğitilen bir akustik model kaynak model olarak kullanılıp az miktarda bir Almanca verisi ile transfer edildiği zaman sadece Almanca'dan eğitilen modelden daha iyi sonuçlar vermektedir [131]. Birden fazla dilin karışımdan oluşan bir veri kümesini kullanıp diller arasındaki ses benzerlikleri öğrenerek çoklu dil tanıyan (multilingual speech recognition) bir akustik model de elde edilebilir [130]. Diller arasındaki bu benzerlikler hem çoklu dil tanıma yapabilen bir konuşma tanıma sistemi için hem de bu modeli kullanarak farklı bir hedef dili tanımak için kullanılabilir.

Transfer öğrenme iki farklı şekilde yapılmaktadır:

- Çoklu görev öğrenme (multi-task learning): Bu yöntemde modelin orta katmanları farklı görevler için ortaklaşa kullanılmaktadır [130, 132, 133, 134]. Modelin son katmanı, her bir görev için ayrı eğitilip o göreve ait çıktılar ve sınıflar bulunmaktadır. Şekil 5.9'da görüldüğü üzere, orta katmanlardaki bilgi birden fazla dil için kullanılıp, modelin son katmanı farklı dillerin fonem setine uygun olarak tasarlanıp eğitilmektedir. Tez kapsamındaki çalışmalarda bu yapı İngilizce ve Türkçe dillerini tek bir modelden tanımak için kullanılmıştır.
- Ağırlık Transferi (weight transfer): Bu yöntemdeki ana fikir, sinir ağlarındaki orta katmanlar ortak bilgileri içerip son katmanların ise probleme özel bilgileri

içermesinden kaynaklanmaktadır. Dolayısıyla, büyük bir veri seti kullanarak önceden eğitilmiş bir model (pre-trained model) elde edip bu modelin son *n* katmanını yeni bir probleme yönelik adapte edebiliriz. Tez çalışmasında, İngilizce konuşmalardan bir model eğitip bu modelin son katmanlarını Türkçe veri ile transfer ederek konuşma tanıma modeli elde edilmiştir.

Bu tezde, transfer öğrenme iki farklı amaç için kullanılmıştır. İngilizce için eğitilen bir akustik model az miktarda Türkçe verisini kullanarak transfer edilip modelin başarı oranı incelenmiştir. Ayrıca, Türkçe için eğitilen bir kaynak model, eğitim setinde bulunmayan farklı bir ortamın verisi ile de transfer edilip bu modelin doğruluk oranı verilmiştir.



Şekil 5.9 Birden fazla dilin verisi ile eğitilen modelin orta katmanları ortaklaşa kullanılıp son katman her bir dil için ayrı eğitilmektedir.

5.6.1. Dile Transfer

Türkçe konuşma tanıma sistemlerinin doğruluk oranı İngilizce'ye göre düşük olmasının sebeplerinden birisi bu dile ait eğitim verisinin az olmasıdır. Farklı aksanlar, ortamlar ve kanallardan gelen konuşmaların sistem tarafında tanınması için bu tür verilerin daha önce eğitim setinde bulunup model eğitiminde kullanılması gerekmektedir. Ayrıca, derin sinir ağlarının yeteri kadar iyi sonuç vermesi için de ihtiyacımız olan en önemli şeylerden birisi büyük miktarda veridir.

Tezin bu kısmında, İngilizce için yapılan akustik modeli kullanarak farklı miktarlarda Türkçe veri ile transfer edip başarı elde edilen modelin kelime hata oranı incelenmiştir. Yöntem olarak ağırlık transferi yöntemi kullanıp İngilizce modelinin orta katmanları genel bilgi olarak kullanılacaktır. Amacımız, İngilizce için bulunan çok miktarda konuşma verisini kullanarak genel konuşma yapısını öğrenebilen bir model elde edip daha sonra bu modeli Türkçe konuşmalara yönelik adapte etmektir. İngilizce akustik model eğitimi LibriSpeech [135] derlemini kullanarak yapılmıştır. Derlemdeki verinin 960 saatlik bir kısmı kaynak model eğitimi için ve 40 saatlik bir bölütü de test için kullanılmıştır (Çizelge 5.10). Kaynak modeli eğitiminden sonra, Türkçe verinin sırasıyla 5, 10, 30 ve 100 saatlik kısmı bu kaynak modelin Türkçe'ye transferi için kullanılmıştır. Türkçe'deki bu veri miktarı tek başına bir akustik model eğitmek için yetersiz olup test sonuçlarından da anlaşıldığı gibi kelime hata oranları yüksektir.

Çizelge 5.10 Kaynak model eğitimi için kullanılan LibriSpeech ve transfer için kullanılacak olan Türkçe veri setlerinin istatistiği.

Train Set	Type	Train (hr)
Libri-Train	Clean	960
Libri-Test	Clean	40
Turkish-Train(1)	Clean	5
Turkish-Train(2)	Clean	10
Turkish-Train(3)	Clean	30
Turkish-Train(4)	Clean	100
Turkish -Test	Clean-Noisy	2

Kaynak model eğitimi Libri-Train veri kümesindeki 960 saatlik ve gürültüsüz konuşmalardan yapılmıştır. HMM/DNN ve DOY (end-to-end) olarak iki farklı yöntem ile kaynak model eğitimi yapılıp Türkçe verisi ile transfer edilmiştir.

HMM/DNN denemelerinde akustik model için 7 katmanlı bir LSTM modeli eğitilmiştir [1]. LSTM'deki ünite sayısı 512 olup aktivasyon fonksiyonu olarak *tanh* fonksiyonu kullanılmıştır. Eğitimin ilk iki iterasyonunda öğrenme oranı (learning rate) 0.001 olarak seçilip, son iki iterasyonunda ise 0.0005 olarak toplamda dört iterasyonlu bir eğitim sonunda SourceDNN olarak kaydedildi. Modelin son katmanındaki çıktılar İngilizce sözlükteki 30 farklı fonemin olasılığını vermektedir. Çizelge 5.11'de LibriTrain verisi ile eğitilen HMM/DNN tabanlı sistemin başarısı LibriTest seti ile denenip kelime hata oranı verilmiştir.

İkinci model eğitimi DOY (End-to-End) model yapısı ile yapılıp SourceE2E olarak kaydedilmiştir. Modelin kodlayıcı (Encoder) tarafında 7 katmanlı bir RNN-LSTM bulunup LSTM'lerin ünite sayısı 512 olarak belirlendi. Çözücü (Decoder) tarafında da yine 7 katmanlı ve 512 boyutlu RNN-LSTM ağı bulunup modelin toplam parametre sayısı SourceE2E ile aynıdır.

LibriTrain derlemindeki 960 saatlik ses dosyaları uzunluklarına göre sıralanıp kısadan uzuna doğru model eğitimine dahil edilmiştir. Model eğitimini kısa dizilerden başlayıp uzun dizilere doğru ilerletmek *SortaGrad* yöntemi olarak bilinip diziden diziye model eğitimlerinin daha stabil bir şekilde devam etmesine yol açmaktadır [70, 106, 136]. Öğrenme oranı ilk iki iterasyon için 0.01 sonraki iki iterasyon için 0.001 ve son altı iterasyonda 0.0005 olmak üzere toplamda 10 iterasyondan oluşan bir eğitim yapılmıştır. SourceDNN ve SourceE2E model eğitimlerinde 100 boyutlu I-vector'ler öznitelik vektörü olarak her 10ms'lik çerçeveler için hesaplanmıştır.

Çizelge 5.11'de LibriTrain derlemini kullanarak HMM/DNN ve DOY modellerinin özellikleri ve LibriTest ile yapılan deneylerin sonuçları raporlanmıştır. HMM/DNN modelindeki kelime hata oranı %11'iken End2End modelinin yaptığı kelime hata oranı %5.29 olarak çıkmıştır. End2End modelindeki hata oranı DeepSpeech2 [70] çalışmasının sonuçları ile karşılaştırabilir olması açısından bu sonuçlar da çizelgenin son satırında verilmiştir.

Çizelge 5.11 LibriSpeech verisi ile eğitilen DOY ve HMM/DNN tabanlı modellerin kelime hata oranı.

Model	#Parameter	Train	Test	%WER
SourceE2E(Libri-Train)	38 M	960 h	40 h	%11
SourceDNN(Libri-Train)	38 M	960 h	40 h	%5.29
DeepSpeech2 **[70]**(Libri-Train)	38 M	960 h	40 h	%5.15

LibriTrain eğitim setinden elde edilen model, kaynak model olarak kullanılıp Türkçe konuşma verisi ile transfer edilecek. Transfer aşamasında, kaynak modelin son *n* katmanı çıkartılıp yerine yeni katmanlar eklenmektedir. Diğer katmanlar da olduğu gibi transfer edilip yeni modelde kullanılacaktır. Kaynak modeldeki son katman İngilizce'ye ait fonemleri içerdiği için bu katmanın mutlaka modelden çıkartılıp Türkçe'deki fonem setine uygun katmanın eklenmesi gerekmektedir. Sondaki diğer katmanlar ise daha çok İngilizce'ye yönelik özellikleri öğrenmiş olup diller arası genel bir bilgi içermemektedir.

Kaynak modeller (SourceDNN, SourceE2E) üzerinde transfer öğrenmeyi uygulamak için bu modellerin ilk 5 katmanı olduğu gibi transfer edilip son iki katman sıfırdan başlatılan yeni iki katman ile değiştirilmiştir. Daha sonra, Türkçe'deki veri ile model eğitimi başlatılıp yeni eklenen katmanlar bu veriye yönelik eğitilmiştir. Eğitim sırasında transfer edilen 5 katman α=0.00001 öğrenme oranı ve yeni eklenen iki katman ise α=0.25 öğrenme oranı ile eğitilmektedir. İlk 5 katman için öğrenme oranını düşük tutmamızın sebebi, bu katmanlardaki parametreler çok fazla etkilenip bilgi kaybı yaşanmadan, yeni eklenen katmanların hızlı bir şekilde eğitilmesidir. Çizelge 5.12'de Türkçe'deki farklı eğitim verilerinden HMM/DNN ve End2End olmak üzere iki ayrı baz model eğitimleri yapılıp transfer model ile karşılaştırma yapılmıştır. Bu çizelgede, iki kaynak modeli üzerinde transfer öğrenme uygulayıp TargetDNN ve TargetE2E olarak adlandırılmıştır.

Sonuçlara göre, İngilizce'den transfer edilen modellerin kelime hata oranı baz modellere göre çok daha düşük olup eğitim verisi arttıkça sonuçlar iyileşmektedir. Turkish-Train(3) verisi ile eğitilen HMM/DNN ve End2End modellerinin kelime hata oranları sırası ile %30.89 ve %31.19'iken aynı veri ile transfer edilmiş modellerin hata oranı %16.00 ve %15.18 olarak elde edilmiştir. Demek ki, 30 saatlik Türkçe veri tek başına model eğitmek için yeterli olmayıp ancak transfer modeldeki bilgileri kullanarak sadece iki yeni katmanın eğitilmesi çok daha etkili olmuştur. Diğer

taraftan da, çizelgenin son satırındaki sonuçlar 100 saatlik bir eğitim verisi ile eğitilen baz modellerin ve transfer edilmiş modellerin sonuçlarını açıklamaktadır. Bu deneyde, baz modelin kelime hata oranı transfer modele göre hem HMM/DNN hem de End2End modelleri için daha düşük çıkmıştır. Bunun nedeni de, baz modelde kullanılan veri miktarı tek başına bir model eğitmek için yeterli olup Türkçe'deki konuşma yapısını transfer modele göre daha iyi modellemesidir.

Turkish-Train(3) verisi ile eğitilen transfer modelin hata oranı Turkish-Train(4) verisinden eğitilen baz modellerine oldukça yakın çıkmıştır. Bu sonuca göre, 30 saatlik Türkçe veri ile İngilizce kaynak model transfer edildikten sonra bu modelin doğruluk oranı 100 saatlik Türkçe veriden eğitilen baz model ile çok yakın çıkmaktadır. Transfer öğrenmedeki amaçlardan birisi de önceden eğitilmiş iyi bir modeli (pre-trained model) az miktarda veri kullanarak başka bir amaca yönelik adapte etmektir.

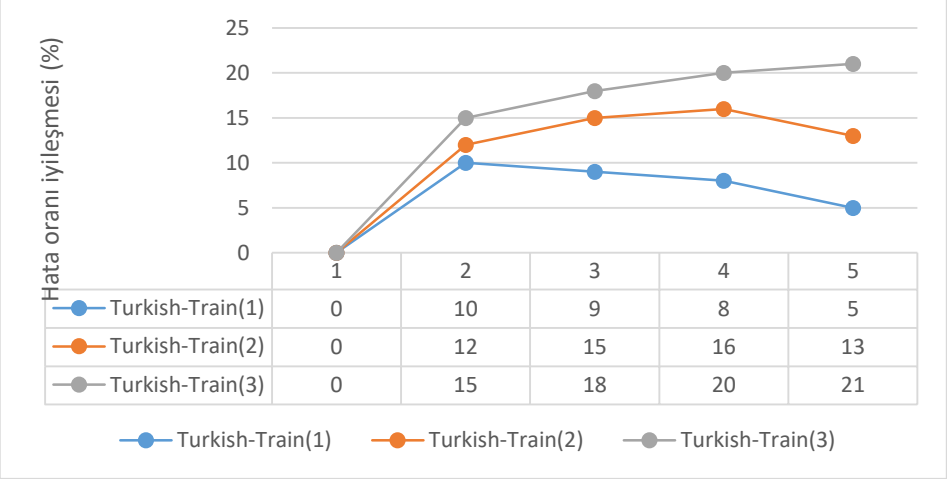
Elde edilen sonuçlara göre, transfer öğrenme yöntemi hedef görev için veri miktarı az olan durumlarda sonuçları daha fazla etkilemektedir. Demek ki, veri hacmi yeterli olduğu durumlarda sadece bu veriden bir model eğitmek önceden eğitilmiş bir model üzerine transfer yapmaktan sonuçları daha fazla pozitif yönde etkiliyor. Bu durumu Çizelge 5.12'deki Turkish-Train(4) verisi ile transfer edilen modelin deney sonuçlarına bakarak da teyit edebiliriz. HMM/DNN ve end2end baz modellerindeki kelime hata oranı sırasıyla %15.45 ve %14.89'iken İngilizce modelleri kullanarak transfer edilen modellerin hata oranı %18.95 ve %17.57'ye yükselmiştir. Bunun nedeni de, bu eğitim setinde bulunan 100 saatlik konuşmanın tek başına bir Türkçe akustik model eğitmeye yeterli olmasıdır.

Transfer eğitiminde, kaynak modelin son 2 katmanı iki yeni katman ile değiştirilip eklenen yeni katmanlar Türkçe veri ile eğitilmiştir. Ancak, değiştirilen katman sayısının sonuçlara olan etkisini de merak ediyoruz. Bu durumu araştırmak için, kaynak modeldeki son 2, 3, 4 ve 5 katmanı yeni başlatılan katmanlar ile değiştirip model eğitimleri yapılmıştır. Şekil 5.10'da İngilizce kaynak modelindeki son *n* katmanı yeni katmanlar ile değiştirip Turkish-Train veri kümeleri ile transfer öğrenmenin etkisi verilmiştir. Turkish-Train(1) verisinde kaynak modeldeki iki katmandan fazla değiştirildiği zaman kelime hata oranı artmaya başlıyor. Bunun nedeni de, bu veri kümesindeki veri miktarının ikiden fazla katmanın eğitilmesi için

yetersiz kalmasıdır. Turkish-Train(2) verisi için ise dördüncü katmandan sonra doğruluk oranı düşmeye başlamıştır.

Çizelge 5.12 Türkçe veri ile eğitilen HMM/DNN ve End2End tabanlı baz modellerin doğruluk oranı İngilizce'den transfer edilmiş model ile karşılaştırılmıştır.

Train	Test	%WER			
		Normal		Transfered	
		HMM/DNN	End2End	TargetDNN	TargetE2E
Turkish-Train(1)	2	45.56	48.26	26.45	40.12
Turkish-Train(2)	2	40.12	43.96	22.17	39.58
Turkish-Train(3)	2	30.89	31.19	**16.00**	**15.18**
Turkish-Train(4)	2	**15.45**	**14.89**	18.95	17.57



Şekil 5.10 Kaynak modelden çıkartılıp sıfırdan başlatılan katman sayısının doğruluk oranına olan etkisi.

5.6.2. Hedef Ortama Transfer

Mevcut bir modeli farklı bir ortamın verisine adapte ederek o ortamdaki konuşma tanıma oranı artırılabilir [137]. Konuşma tanıma motoruna gönderilen sesler sabit bir cihazın mikrofondan geliyorsa ve bu tür ses örnekleri model eğitimine dahil edilmemişse sistemin doğruluk oranı düşüyor. Bu durumlarda bu mikrofona ait az miktarda bir veriyle baz model üzerinde transfer öğrenme yapıp modeli bu mikrofona yönelik adapte edebiliriz. Örneğin, Amazon Alexa gibi akıllı asistanlar cihaz

içerisinde bulunan sabit bir mikrofonla alınan ses kayıtlarını konuşma tanıma motoruna gönderiyorlar. Dolayısıyla, kullanılan model bu mikrofona ve kanal özelliklerine yönelik adapte edilirse sistemin doğruluk oranı artırılabilir.

Bu durumu araştırmak için tek bir cep telefonu ile farklı ortamlarda kaydedilen ve 1000 cümleden oluşan bir veri kümesi oluşturuldu. Toplanan ses kayıtları ev, iş yeri, alış veriş merkezi ve cadde gibi çeşitli ortamlardan alınmıştır. Bu verinin 800 cümlelik bir kısmı transfer verisi olarak kullanılıp geriye kalan 200 cümle ise deney için ayrılmıştır. Bu veri setindeki kelime sayısı 5689 tane olup tekil kelime sayısı 1236 tanedir.

Kaynak model eğitimi veri toplama bölümünde özellikleri verilen 382 saatlik Türkçe konuşma ile eğitilmiştir. Model eğitiminde DOY yapısı kullanılıp 7 katmanlı bir RNN-LSTM kodlayıcı-çözücü yapısı bulunmaktadır. Hem kodlayıcı tarafındaki hem de çözücü tarafındaki son LSTM katmanları modelden çıkartılıp yeni başlatılan katman ile değiştirilmiştir. Transfer model eğitimi 800 cümlelik ve yaklaşık 50 dakikalık bir veri kümesi ile yapılmıştır. Eğitim aşamasında, kaynak modelden alınan 6 katmanın öğrenme oranı 0.0001 olarak tutulup eklenen yeni katman için ise 0.25 olarak sabitlenmiştir. Bunun nedeni de, kaynak modeldeki katmanlar çok fazla etkilenmeden yeni eklenen katmanın hedef ortamına yönelik eğitilmesidir.

Kaynak model ve transfer model ile 200 cümlelik deney setindeki sonuçlar Çizelge 5.13'te verilmiştir. Kaynak modelin bu test kümesindeki kelime hata oranı %20.45'iken transfer modeli kullanarak hata oranı %8 olarak çıkıp relatif olarak %56'lık bir iyileştirme sağlanmıştır.

Çizelge 5.13 Kaynak model olan ve 382 saat veriyle eğitilen modelin doğruluk oranı ile hedef ortama yönelik transfer edilen modelin doğruluk oranları.

Model	#Sentence	Vocab Size	WER(%)	Transfered Layer
Tr 382 h	200	248	%20.45	-
Transfer	200	248	%8.98	2

5.7. Noktalama İşaretleri

Konuşma tanıma sistemleri genelde noktalama işaretleri içermeyen ham bir çıktı üretiyorlar. Noktalama işaretleri ve formatlama gibi kurallar metinlerin insan tarafından rahat okunması ve gramatikal olarak doğruluğu açısından oldukça önemlidir. Ayrıca, konuşma tanıma çıktısını kullanan doğal dil işleme, makine çevirisi ve metin özetleme gibi farklı sistemler de noktalama işaretlerinin bulunduğu bir metinde daha başarılı sonuçlar vermekteler [138, 139]. Tezin bu kısmında, RNN-LSTM modelini kullanarak ham bir metindeki noktalama işaretlerinin restore edilmesi için yaptığımız çalışmalar anlatılacaktır. Geri döndürülmesini istediğimiz noktalama işaretleri !_ünlem ?_soru-işareti ._nokta ve ;_noktalı-virgül'dür.

Noktalama işaretlerini restore etmek için literatürde farkı yöntemler bulunmaktadır. Bazı çalışmalarda sadece metin verisi kullanılırken bazılarında ise konuşmadaki ek bilgiler de kullanılıyor. N-gram tabanlı çalışmalarda [140] noktalama işaretleri bulunan bir metini kullanarak n-gram istatistikleri hesaplanıp bu değerlere metindeki noktalamalar tahmin ediliyor. Conditional random Field (CRF) [141], Transition Based Dependecy Parsing [142] gibi yöntemler de noktalama işaretlerinin restore edilmesi için kullanılıyor. Bazı çalışmalarda da [139, 143] noktalama işaretlerini döndürmeyi bir makine çevirisi problemi olarak ele alıp ham metini noktalama işareti olan bir metine çeviren modeller kullanılmıştır. Konuşmadaki kelimeler arasındaki duraksamaları da ek bilgi olarak kullanan çalışmalarda sonuçlar daha da iyileşmiştir [144, 145].

Tez kapsamındaki çalışmada, sadece metin bilgisi kullanarak Türkçe konuşma tanıma çıktısında noktalama işaretleri restore eden bir model çalışması yapılmıştır. Model olarak RNN-LSTM yapısı kullanılıp ham metin kodlandıktan sonra çözücü bir ağ vasıtasıyla noktalama işaretleri döndürülüyor. RNN içerisindeki LSTM üniteleri, cümledeki kelimelerin birbirine olan anlamsal ve yapısal bağımlılıklarını öğreniyor. Ayrıca, noktalama işaretlerinin eklenmesi için daha fazla önem taşıyan kelimelerin model tarafından öğrenilmesi için odaklanma mekanizması eklenmiştir [92]. Örneğin, soru işareti eklemek için "*mı, mi, mu, ve mü"* gibi sözcüklerin daha önemli olduğu odaklanma mekanizması tarafından öğrenilmektedir.

5.7.1. Eğitim Verisi

Eğitim verisi olarak, Türkiye Büyük Millet Meclisi (TBMM) tutanaklarındaki metinler kullanıldı. Bu metinlerde noktalama işaretleri son derece özenli bir şekilde yerleştirilmiş olup model eğitimi ve deneylerin doğru yapılabilmesi için oldukça uygundur. Tutanaklar toplu bir şekilde TBMM sayfasından çekilip birtakım ön işlemeler uygulandıktan sonra hazır bir metin derlemi haline getirildi. TBMM'de stenograflar tarafından yazılan ve düzenlemeler yapıldıktan sonra yayınlanan bir tutanağın örneği Şekil 5.11'de verilmiştir.

Tutanaklar genelde paragraflar halinde hazırlanıp her paragrafta ortalama 4.2 cümle ve 61 kelime bulunmaktadır. Eğitim aşamasında, paragraftaki kelime dizileri kodlayıcı tarafından işlenip çözücü tarafı ise gerekli olan yerlerde noktalama işaretlerini üretmektedir. Model eğitimindeki dizilerin uzunluğu ortalama 61 kelime olduğu için test aşamasında ise konuşma tanıma çıktısını 61'lik kelime grupları halinde bu sisteme veriyoruz.

Değerli arkadaşlar, Adalet ve Kalkınma Partisinin on dört yıllık iktidarı boyunca en büyük sorun yaşadığımız yerlerden bir tanesidir dış politika. Dış politikada farklı kavramlarla beraber başladı Adalet ve Kalkınma Partisi. Öncelikle "Komşularla sıfır sorun" diyerek başladı ki Sayın İsmail Cem'in döneminde temelleri atılmış bir siyasetti, doğru bir siyasetti. Adalet ve Kalkınma Partisinin bunu ilk başlarda takip ediyor olması, devam ettiriyor olması da olumlu bir gelişmeydi fakat bu çok hızlı bir şekilde "Lider ülke Türkiye"ye döndü, "Türkiye her şeye kadir. Orta Doğu'da Türkiye'nin dışında hesap yapılamaz. Dolayısıyla Türkiye, Orta Doğu'da ve kendi coğrafyasında her şeyi belirleyen ülkedir." noktasına doğru bir dönüş gerçekleşti. Fiyaskoyla sonuçlandı. Döndük, elimizdeki fiyaskoyu "Değerli yalnızlık" adı altında ulvileştirdik. "Aslında biz çok ilkeli bir siyaset izledik ama dünya bizi anlamadı, dolayısıyla yalnız kaldık." siyasetine döndük. Ondan sonra, şu anda da tekrardan olumlu bir noktaya doğru en azından teorik olarak bir gidişat var. Ne diyor Sayın Başbakan? Diyor ki: "Dostlarımızı artıracağız, düşmanlarımızı azaltacağız." Birazdan, konuşmanın sonlarına doğru, geçen hafta Amerika'daydık Dışişleri Komisyonu Heyeti olarak, onunla ilgili de bazı bilgiler veririm. Ama şunu da görmek lazım: "Dostlarımızı artıracağız, düşmanlarımızı azaltacağız." demeniz için zaten düşmanlarımızın sayısının çok fazla yükselmiş olması lazım mantıken, doğrusu da o. Şöyle bir bakarsanız dünyadaki genel gidişata, bir yandan Türkiye sürekli Amerika'yla bir gerilim siyaseti izliyor bir süredir; Suriye'de çıkarlarımız çakışıyor, bazı yerlerde örtüşüyor, bazı yerlerde örtüşmüyor. Rusya'yla şu anda -Cumhurbaşkanı da Rusya'daydı- bazı ilerlemeler kaydedildi politik konularda. Şimdi, biz bu noktaya niye geldik? Bir sene önce Rusya'yla biz zaten dosttuk, ne değişti de biz Rusya'yla kavga ettik ve daha sonra bu noktaya gelip kaybettiğimizi tekrardan bulmaya sevinir hâle geldik?

Başka bir durum: İsrail bizim Orta Doğu'da eskiden beri köklü ilişkilerimiz olan bir ülkeydi ama siyasi olarak zaten reddettiği, ilişki kurmak istemediği bir siyasi çizgiden geliyor Adalet ve Kalkınma Partisi. Döndük hızlı bir şekilde altı senede ilişkilerimizi dibe vurdurduk, ondan sonra aynı ilişkiyi tekrardan kurabilmek için de müthiş bir çaba gösterdik.

Benzer durumu biz Avrupa Birliğiyle ilgili yaşıyoruz. Bir vize muafiyeti meselesi gündeme geldi, bütün komisyonlarımız çalıştı, biz Dışişleri Komisyonu olarak ciddi mesai yaptık bu 72 kriter gerçekleşsin diye ama ortaya çıkan sonuç, Türkiye'nin Avrupa Birliğiyle vize muafiyeti meselesi sırf önceki Başbakan Davutoğlu bu konuyu istedi ve bu konudan prim yapacak diye rafa kaldırıldı. Yani, biz döndük Türkiye'nin dış politikasıyla ilgili birçok meseleyi kendi ülkemizin iç politikasına alet ettik, dışarıdaki liderlere kendi meydanlarımızda siyasi parti liderleri olarak salvo yaptık; hatta daha ileriye gittik, bazı konularda, özellikle Avrupa Birliği vize muafiyeti konusunda meseleyi aldık, Türkiye'nin iç malzemesi hâline getirmenin ötesine geçtik, Adalet ve Kalkınma Partisinin kendi iç meselesi hâline getirdik yani çağ atladık aslında bu konuda.

Şimdi, arkadaşlar, dünyanın genel gidişatına, Türkiye'nin dış politikasına baktığımız zaman -on dört yıllık iktidarı boyunca Adalet ve Kalkınma Partisinin- toplamına baktığımızda açıkçası hiç kimse bir başarıdan, müthiş bir gelişmeden bahsedemez. Aşama aşama iyi yapılmış şeyler var, tabii ki var, onun için teşekkür ediyoruz. Biz, yapılamayan işler için eleştiriyoruz ve diğer muhalefet partileri de dâhil olmak üzere ama özellikle Cumhuriyet Halk Partisi bütün bu yanlışları dile getiriyor olmasına rağmen şimdiye kadar bunlar kale alınmadı ve Türkiye gerek ekonomik açıdan sıkıntı yaşadı... Bilirsiniz, neredeyse son bir buçuk yıldan beri ihracatımız istikrarlı bir şekilde düşüyor. Suriye'de yaşamış olduğumuz bu sorunun, Irak'ta yaşamış olduğumuz problemlerin Rusya'yla, İsrail'le, Avrupa Birliği'yle, Amerika'yla, birçok ülkeyle yaşamış olduğumuz gerilimlerin sonucunda dış politika bizim ekonomimize zarar verir hâle geldi. Normalde dış politika, bizim ekonomimizde dış ticaretimizi genişleten bir süreç izlemesi gerekirken, böyle bir katkı sunması gerekirken tam tersi bir durum izlemeye başladı.

Bir başka nokta: Bizim Kilis ilimiz, Suriye'de izlemiş olduğumuz yanlış politikanın sonucunda cumhuriyet tarihi boyunca ilk defa füzelerle vuruldu arkadaşlar. Bizim bir ilimiz sürekli füzelerle vurulan bir yere dönüştü. "Emevi Camisi'nde namaz kılacağız." diye yola çıkan bir siyaset izlendi, Kilis'te Ulu Cami'de namaz kılamaz hâle geldik, başımıza sürekli füzeler yağıyordu. Şimdi, böyle bir siyaseti devam ettirebilmek, böyle bir siyaseti başarı diye sunmak, böyle bir siyaseti hiçbir şey olmamış gibi inatla, muhalefetin sözünü dinlemeden, muhalefetin eleştirilerine kulak asmadan devam ettirmek, Türkiye'nin gelecekte, önümüzdeki yıllarda da, bugün de içinde bulunduğu duruma hiçbir katkı sağlamaz.

Suriye'yle ilgili, örneğin, resmen bazı çetelere lojistik destek verdik, bazı örgütlere destekler verdik. O örgütlerin kimler olduğu... İsim değiştiriyorlar, tavır değiştiriyorlar, liderlik değiştiriyorlar, kendi aralarındaki ittifakları değiştiriyorlar ve bir süre sonra kimin ılımlı kimin değil, kimin muhalif kimin aslında rejimle iş birliği yapan olduğunun günlük olarak değiştiği bir Suriye'de Türkiye tam anlamıyla sınıfta kalan bir siyaset izlemiş oldu.

Sınırlarımızı biz militan geçişine çok açık bir hâle getirdik özellikle Suriye'de. Kürt sorunu, Türkiye'nin kendi içinde barışla, görüşmelerle çözülmesi gereken bir meseleyken Suriye'de izlemiş olduğumuz yanlış politikanın sonucunda artık Suriye'deki sorunun bir parçası hâline geldi, aslında bir tür bölgeselleşti ve belki uluslararasılaşma yönünde bir mesafe almaya başladık.

Suriye'nin iç savaş çıktığında nüfusu 20 milyonun biraz üzerindeydi. Biz 3 milyon Suriyeli mülteciyi ağırlıyoruz yine Suriye'de izlemiş olduğumuz yanlış politikanın sonucunda. Yani, Suriye nüfusunun yüzde 15'i Türkiye'de arkadaşlar. Bunu bir misafirperverlik olarak sundu Adalet ve Kalkınma Partisi. E, tabii ki, biz, kökleri, gelenekleri itibarıyla misafirperver bir milletiz ama "3 milyon kişi bize misafirliğe niye geldi?" diye sorma hakkımızı kullanmadık. Bunu söylediğimiz zaman, bunu eleştirdiğimiz zaman iktidar partisinden arkadaşların çok da bu işe kulak asmadığını gördük.

Şekil 5.11 Yayınlanmış olan TBMM tutanaklarından bir örnek.

Çizelge 5.14'te, model eğitimi ve deneyler için kullanılan metin derleminin bilgileri verilmiştir. Eğitim setinde yaklaşık 2 milyon paragraf bulunup toplam kelime sayısı 110 milyon tanedir. Noktalama işaretleri olarak ele alacağımız !_ünlem ?_soru-işareti ._nokta ve ;_noktalı-virgül karakter sayısı eğitim setinde 10.8 milyon olup test setinde ise 2.4 milyon tane bulunmaktadır.

Çizelge 5.14 Model eğitimi ve deneyler için kullanılan metin derleminin özellikleri.

Corpus	#Paragraph	#Word	Vocab Size	#Punctuation
Eğitim	2 M	110 M	450 K	10.8 M
Test	500 K	27 M	180 K	2.4 M

RNN modelindeki kodlayıcı taraf noktalama işaretleri bulunmayan bir kelime dizisini girdi olarak alıp işledikten sonra çözücü tarafa iletmektedir. Çözücü taraf da kodlayıcı tarafından gelen bilgi vektörü ile noktalama işaretli metini alıp model parametrelerini güncellemektedir.

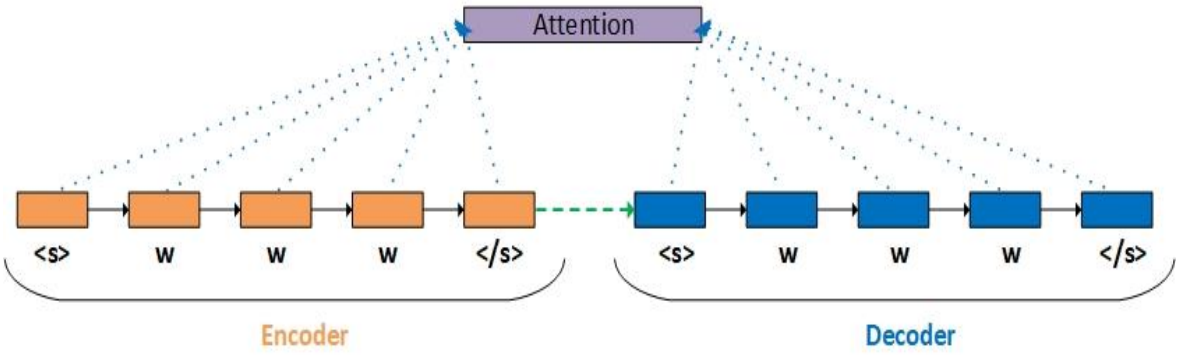
5.7.2. Model

Ham metindeki noktalama işaretlerini restore edecek bir sistem, girdideki bir diziyi başka diziye çevirmesi gerektiğinden dolayı seq2seq sinir ağları yapısına uygundur. Bu model yapısında, girdiler sırayla kodlayıcı tarafa verilip son girdi işlendikten sonra bilgi vektörü ile birlikte çözücü tarafa iletilmektedir. Çözücü taraf bilgi vektörü ile üretilmesi gereken çıktıları alıp modelin parametrelerini bu çıktıları üretebilecek şekilde güncellemektedir.

Kodlayıcı taraf, *t* zamanındaki bir girdiyi alıp RNN'deki katmanlardan geçirdikten sonra üretilen saklı değeri (hidden value) *t+1* zamanına aktarıyor. Girdideki *n* tane sembol işlenene kadar kodlayıcı *t+n* anına kadar devam edip bilgi vektörünü üretmektedir (Şekil 5.12). Modeldeki kodlayıcı girdideki sembolleri sırayla işleyip </s> sembolünü görene kadar devam ediyor. Bilgi vektörü elde edildikten sonra <s> sembolü ile çözücü tetiklenip çıktı dizisini üretiyor. Ayrıca modeldeki odaklanma mekanizması, çıktıların üretim aşamasında hangi girdiler üzerinde daha fazla odaklanması gerektiğini öğrenip bu girdilere daha fazla ağırlık veriyor.

Kodlayıcı ve çözücü tarafında iki katmanlı ve çift taraflı RNN kullanılmıştır (BIRNN). İlk katman girdiyi soldan sağa işleyerek ilerleme (forward) aşamasını gerçekleştirip ikinci katman ise girdiyi tersten işleyerek gerileme (backward) aşamasını

gerçekleştiriyor. Kelimeler one-hot vektörü şeklinde modele verilip modelin daha hızlı eğitilmesi ve kelimeler arasındaki ilişkilerin modellenmesi için girdi katmanına gömme katmanı (Embedding Layer) yerleştirilmiştir [146]. Bu katmanın görevi, seyrek (Sparse) yapısı olan one-hot vektör gösterimini yoğun (Dense) bir girdi vektörüne dönüştürmektir.



Şekil 5.12 Noktalama işaretlerini restore eden diziden diziye bir sinir ağı modeli.

Noktalama işaretleri içermeyen bir metin modelin girdisi olarak kullanılıp noktalama işaretleri içeren hali ise çıktı olarak kullanılacaktır (Şekil 5.13). Seq2seq bir model, bu girdi ve çıktıları kullanarak ham bir metindeki noktalama işaretlerini nasıl restore etmesi gerektiğini öğrenecektir. Girdideki kelimelerin aynısı çıktı olarak da üretilip sadece gerekli yerlerde noktalama işaretlerinin yerleştirilmesi beklenmektedir.

Test sırasında, çözücü taraf kodlayıcı taraftan gelen bilgi vektörü ve ürettiği bir önceki sembolü tüketerek kelimeler ve noktalama işaretlerini üretiyor. Fakat bir kelime yanlış üretildiği zaman, zincir bir şekilde çözücü taraf sonuna kadar yanlış kelime sırası üretebiliyor. Bu sistemin amacı konuşma tanıma çıktısını formatlama olduğu için, girdideki kelimelerin bu model tarafından değiştirilmemesi gerekiyor. Bu durumu engellemek için eğitim sırasında çıktı dizisi olarak kelimeler yerine etiketleri kullanıyoruz. Bu etiketlerin görevi üretilen noktalama işaretlerinin yerini belirleyip hangi kelimeler arasına yerleştirilmesi gerektiğini göstermektir. Ayrıca, üretilen kelimelerin küçük veya büyük harflerle yazılması bu etiketlerle belirleniyor.

Girdideki her kelime çıktı tarafında aşağıdaki etiketler ile gösteriliyor:

- kelime = <word>
- Kelime = <Word>
- KELİME = <WORD>
- Sayı = <Digit>

Girdi:

benzer durumu biz avrupa birliğiyle ilgili yaşıyoruz bir vize muafiyeti meselesi gündeme geldi bütün komisyonlarımız çalıştı biz dışişleri komisyonu olarak ciddi mesai yaptık bu 72 kriter gerçekleşsin diye ama ortaya çıkan sonuç türkiye'nin avrupa birliğiyle vize muafiyeti meselesi sırf önceki başbakan davutoğlu bu konuyu istedi ve bu konudan prim yapacak diye rafa kaldırıldı yani biz döndük türkiye'nin dış politikasıyla ilgili birçok meseleyi kendi ülkemizin iç politikasına alet ettik dışarıdaki liderlere kendi meydanlarımızda siyasi parti liderleri olarak salvo yaptık hatta daha ileriye gittik bazı konularda özellikle avrupa birliği vize muafiyeti konusunda meseleyi aldık türkiye'nin iç malzemesi hâline getirmenin ötesine geçtik adalet ve kalkınma partisinin kendi iç meselesi hâline getirdik yani çağ atladık aslında bu konuda

Çıktı:

Benzer durumu biz Avrupa Birliğiyle ilgili yaşıyoruz. Bir vize muafiyeti meselesi gündeme geldi, bütün komisyonlarımız çalıştı, biz Dışişleri Komisyonu olarak ciddi mesai yaptık bu 72 kriter gerçekleşsin diye ama ortaya çıkan sonuç, Türkiye'nin Avrupa Birliğiyle vize muafiyeti meselesi sırf önceki Başbakan Davutoğlu bu konuyu istedi ve bu konudan prim yapacak diye rafa kaldırıldı. Yani, biz döndük Türkiye'nin dış politikasıyla ilgili birçok meseleyi kendi ülkemizin iç politikasına alet ettik, dışarıdaki liderlere kendi meydanlarımızda siyasi parti liderleri olarak salvo yaptık; hatta daha ileriye gittik, bazı konularda, özellikle Avrupa Birliği vize muafiyeti konusunda meseleyi aldık, Türkiye'nin iç malzemesi hâline getirmenin ötesine geçtik, Adalet ve Kalkınma Partisinin kendi iç meselesi hâline getirdik yani çağ atladık aslında bu konuda.

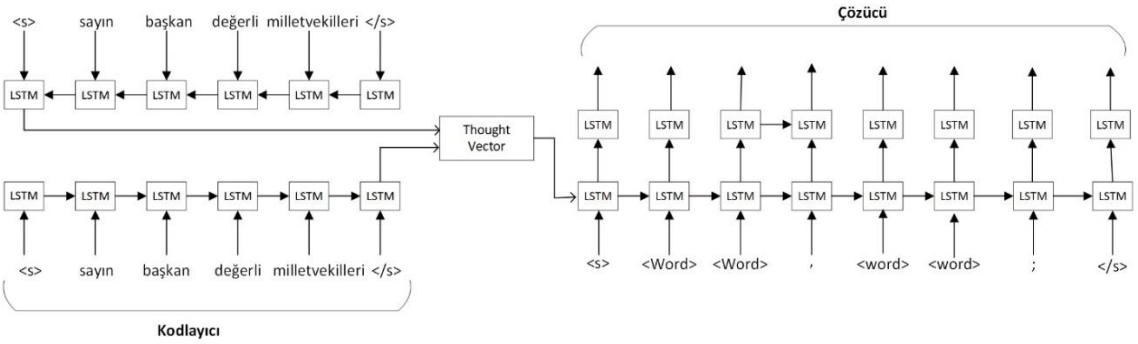
Şekil 5.13 Diziden diziye çeviri yapan bir modele verilecek olan girdi ve çıktı örneği. Girdideki metinde noktalama işaretleri bulunmayıp tüm harfler küçük karakter ile yazılmıştır. Çıktı dizisinde ise noktalama işaretleri bulunup gerekli yerlerde büyük karakterler kullanılmıştır.

Etiket tabanlı çıktı, çözücü tarafının sözlük boyutunu binlerce kelimden dört etikete indirerek hatalı çıktı üretmesini engelliyor. Bu yapıdaki girdi ve çıktının örneği aşağıdaki gibidir:

Girdi: **Benzer durumu biz Avrupa Birliğiyle ilgili yaşıyoruz. Bir vize muafiyeti meselesi gündeme geldi,**

Çıktı: **<Word> <word> <word> <Word> <Word> <word> <word> . <Word> <word> <word> <word> <word> <word> ,**

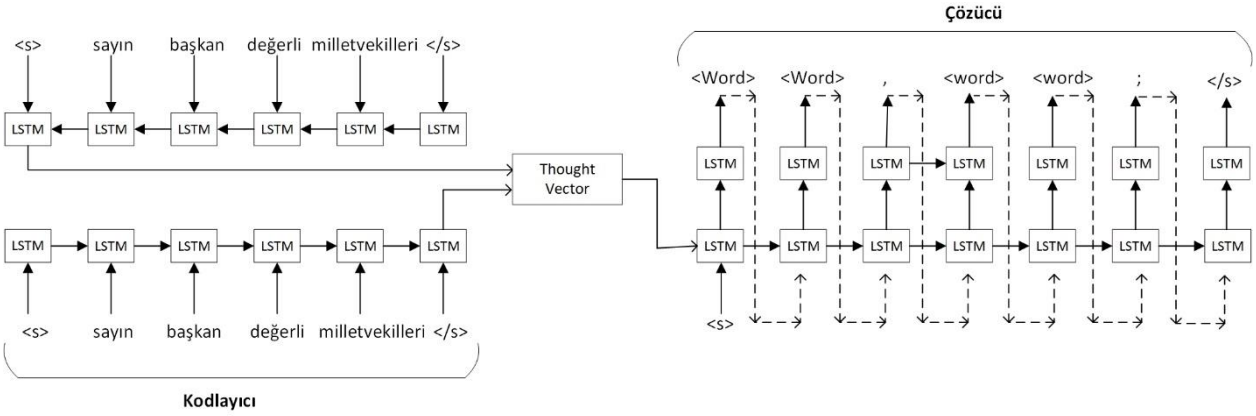
Bu yapıda, girdi dizisi kelime listesi olup çıktılar da etiketler ve bunlar arasındaki noktalama işaretleridir. Model bu girdi/çıktı dizisini kullanarak doğru sayıda etiketi üretip gerekli olan yerlere noktalama işaretlerini yerleştiriyor (Şekil 5.14). Kodlayıcının ilk katmanı, cümleyi soldan sağa işleyerek kelimeleri LSTM ünitelerinden geçirip son kelimeye kadar bu işlemi tekrarlamaktadır. İkinci katman da kelimeleri sağdan sola alıp ilk kelimeye ulaşana kadar devam ediyor. Bu işlem sonucunda, ilk katmandaki son LSTM ve ikinci katmandaki ilk ünitenin saklı değerleri (Hidden Value) birleştirilip bilgi vektörü olarak çözücü tarafına iletilmektedir. Çözücü bu bilgi vektörünü ilk LSTM ünitesini başlangıç değeri olarak alıp doğru çıktıları tüketerek model çıktılarını üretiyor.



Şekil 5.14 Etiket tabanlı model eğitiminde kullanılan RNN-LSTM yapısı

Sistemin test aşamasında, girdi tarafındaki gibi bir cümle verildiği zaman çıktı olarak etiketler üretilmektedir. Elde edilen çıktı basit bir son işlemeye (post-processing) tabi tutulup etiketler girdi tarafındaki kelimeler ile yer değişiyor. Şekil 5.15'te modelim

test aşamasındaki yapı çizilmiştir. Girdide olarak verilen "*sayın başkan değerli milletvekilleri"* kelime dizisi, RNN'in ilk ve ikinci katmanından geçtikten sonra elde edilne bilgi vektörü çözücüye aktarılıyor.



Şekil 5.15 Modelin test aşamasındaki işleyişi. Çözücü kendi ürettiği çıktıları girdi olarak tüketiyor.

Şekil 5.14 ve Şekil 5.15'ten de anlaşıldığı üzere eğitim ve test aşamasındaki modellerin yapısı farklıdır. Aslında, bu iki modelin kodlayıcı tarafı aynı şekilde girdideki kelime sırasını işleyip çözücü tarafa iletmektedir. Eğitim sırasındaki çözücü hep doğru çıktıları tüketip, test aşamasındaki çözücü ise bir önceki ünitenin çıktısını girdi olarak kullanıyor.

Eğitim setindeki yaklaşık 2 milyon paragraf, etiket tabanlı girdi/çıktı yapısına dönüştürüldükten sonra model eğitimi başlatıldı. Eğitimler tek bir GPU kartı üzerinde yapılıp 48 saatlik bir eğitim sonunda noktalama işaretlerini restore eden bir model elde edildi. Doğruluk oranını test etmek için test kümesinde bulunan 500 bin paragraf üzerinde ön işleme uygulanıp test seti oluşturuldu. Elde edilen sonuçlarda Recall, Precision ve F1 skoru gibi parametreler ölçülüp Çizelge 5.15'te raporlanmıştır.

Çizelge 5.15 Üç farklı model ile noktalama işaretlerini restore eden sistemin metrikleri.

Model	**Virgül**			**Nokta**			**Soru**			**Nokta.Virgül**		
	Pr.	Re.	F1	Pr.	Re.	F1	Pr.	Re.	F1	Pr.	Re.	F1
RNN-LSTM	52.1	30.5	38.4	45.8	32.7	38.1	55.8	39.8	46.4	40.2	29.5	34.0
RNN-BLSTM	61.9	41.8	49.9	55.7	38.4	45.4	59.5	40.2	47.9	42.5	30.2	35.0
RNN-BLSTM-Att	**63.4**	**58.2**	**60.6**	**68.9**	**55.8**	**61.6**	**82.5**	**70.9**	**76.2**	**52.8**	**40.5**	**45.8**

Sonuçlara göre soru işareti diğer noktalama işaretlerine göre daha yüksek bir oranda modeller tarafından doğru tahmin edilmiştir. Bunun nedeni de Türkçe'deki soru eklerinden sonra soru işareti gelmesi gerektiğinin modeller tarafından öğrenmiş olmasıdır.

SONUÇ

Bu tez kapsamında uçtan uca eğitilen bir konuşma tanıma modelinin yapısı anlatılmıştır. Bu model klasik bir konuşma tanıma sistemi içerisinde bulunması gereken bileşenleri kendi içerisinde barındırıp tek bir model olarak eğitilmektedir. Model eğitiminde kullanılan girdiler ve çıktılar arasında herhangi bir bağımsızlık varsayımı bulunmayıp, çıktılardaki uzun bağımlılıklar model tarafından öğrenilmektedir. HMM tabanlı sistemler de uçtan uca eğitilebilir ancak bu modellerde güçlü bir dil modeli kullanılmazsa modelin kelime hata oranı oldukça yüksek çıkmaktadır. Bu tezde kullanılan model hem sesteki akustik bilgiyi hem de dildeki yapısal bilgiyi tek bir modelde öğrendiği için dil modeli kullanılmadığı durumlarda bile yüksek doğruluk oranı elde edilmektedir.

Gözetimli makine öğrenimindeki (supervised machine learning) en temel adımlardan birisi yeterince etiketlenmiş verinin toplanıp güçlü bir modelin bu veriyle eğitilmesidir. Konuşma tanımada da doğruluk oranı yüksek bir akustik model, binlerce saat yazıya dökülmüş ve çeşitli ses örneklerini içeren derlemler ile eğitilmektedir. Türkçe için hazır bulunan ve akademik çalışmalarda kullanılabilecek bu derlemlerin büyüklüğü oldukça az olup İngilizce ile karşılaştırılabilir değildir. Bu tez çalışmasında 56 saatlik Türkçe konuşma dosyaları farklı kaynaklardan toplanıp el yordamı ile yazıya döküldükten sonra derlem olarak hazırlanmıştır.

Derin sinir ağlarının konuşma tanımda kullanılması bu sistemlerin doğruluk oranını oldukça artırmıştır. Klasik HMM tabanlı sistemlerde akustik model olarak sinir ağını kullanmak GMM'e göre daha iyi sonuçlar verip kelime hata oranını düşürmüştür. Bu sistemlerdeki HMM yapısını RNN ile değiştirdiğimiz zaman hem sesteki dinamikliği hem de seslerin akustik modelini tek bir modelde öğrenebiliriz. Tez çalışmasında önerilen model RNN-LSTM yapısını kullanıp dinamik uzunluktaki bir ses girdisini farklı uzunlukta olan bir karakter veya kelime dizisine çevirmektedir. Odaklanma mekanizmasını kullanarak da, hem çıktılarla girdiler arasındaki bağımlılık derecesini kontrol edip hem de ses ile yazı arasındaki zaman hizalaması elde edilmiştir. Ayrıca, çıktı olarak Türkçe'deki kelimeler yerine kök-ek tabanlı bir yöntem kullanarak modelin kelime hata oranı biraz daha düşürülmüştür. Türkçe gibi kısıtlı kaynaklı diller için transfer öğrenme yöntemi uçtan uca mimaride denenip başarılı sonuçlar elde edilmiştir.

Konuşma tanıma çıktısındaki ham metini formatlayıp noktalama işaretlerinin yerleştirilmesi için RNN-LSTM tabanlı bir diziden diziye sinir ağı modeli eğitilmiştir. Bu model ham bir metini alıp RNN'deki kodlayıcı-çözücü yapısı ile noktalama işaretlerinin yerleştirilmesi gereken yerleri belirlemektedir. Ayrıca özel isimler ve kısaltmalar gibi kelimelerde etiket tabanlı yöntem sayesinde doğru olarak yazılıp daha anlamlı bir çıktı elde edilmiştir. HMM tabanlı konuşma tanıma sistemlerinde okunuş sözlüğü kullanıldığı için kelimelerin farklı okunuşları sözlükte bulunmalıdır. Türkçe kelimelerde olası farklı okunuşları ve İngilizce kelimelerin Türkçe okunuşlarını üreten bir RNN modeli de bu tez kapsamında eğitilmiştir. Bu model farklı okunuşları üretebildiği için sistemin kelime hata oranını biraz daha düşürmüş ve özellikle yabancı kelimelerin okunuşunu elde etmek için kullanılmıştır.

Gelecekteki çalışmalarda, modeldeki tüm adımların uçtan uca olması ve minimum müdahile ile bir model eğitilmesi için öznitelik çıkarma adımları da ortadan kaldırılıp ham ses sinyalini kullanarak model eğitimi yapılabilir. Model eğitiminde de daha fazla ses verisi kullanarak, bu modelin başarısı hem akustik modelleme için hem de kelimelerin istatistiksel yapısını modellemesi açısından artırılabilir. Bu çalışmada sonuçların yeniden skorlanması *n-gram* tabanlı bir dil modeli kullanılarak yapılmıştır. Dil modelinin de tekrarlanan bir sinir ağı ile eğitilmesi ve tek bir model olarak akustik modele entegre edilmesi sistemin yapısını biraz daha basitleştirecektir.

KAYNAKLAR

[1] Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., and Schwarz, P. The Kaldi speech recognition toolkit. in *IEEE 2011 workshop on automatic speech recognition and understanding*. **2011**. IEEE Signal Processing Society.

[2] Arisoy, E., Can, D., Parlak, S., Sak, H., and Saraçlar, M., Turkish broadcast news transcription and retrieval. IEEE Transactions on Audio, Speech, and Language Processing, **2009**. 17(5): p. 874-883.

[3] Salor, Ö., Pellom, B. L., Ciloglu, T., Hacioglu, K., and Demirekler, M. On developing new text and audio corpora and speech recognition tools for the turkish language. in *INTERSPEECH*. **2002**.

[4] Manning, C. D. and Schütze, H., *Foundations of statistical natural language processing*. Vol. 999. **1999**: MIT Press.

[5] Young, S., Evermann, G., Gales, M., Hain, T., Kershaw, D., Liu, X., Moore, G., Odell, J., Ollason, D., and Povey, D., *The HTK book.* Cambridge university engineering department, **2002**. 3: p. 175.

[6] Morgan, N. and Bourlard, H. Continuous speech recognition using multilayer perceptrons with hidden Markov models. in *Acoustics, Speech, and Signal Processing, 1990. ICASSP-90., 1990 International Conference on*. **1990**. IEEE.

[7] Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A.-r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., and Sainath, T. N., Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine,* **2012**. 29(6): p. 82-97.

[8] Lenzo, K., *The cmu pronouncing dictionary*. **2007**.

[9] Mikolov, T., Karafiát, M., Burget, L., Cernocký, J., and Khudanpur, S. Recurrent neural network based language model. *in Interspeech.* **2010**.

[10] Povey, D., Discriminative training for large vocabulary speech recognition. **2005**, University of Cambridge.

[11] Veselý, K., Ghoshal, A., Burget, L., and Povey, D. Sequence-discriminative training of deep neural networks. *in Interspeech.* **2013**.

[12] Graves, A. and Jaitly, N. Towards end-to-end speech recognition with recurrent neural networks. in *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*. **2014**.

[13] Katz, S., Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE transactions on acoustics, speech, and signal processing,* **1987**. 35(3): p. 400-401.

[14] Kneser, R. and Ney, H. Improved backing-off for m-gram language modeling. in *Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 International Conference on*. **1995**. IEEE.

[15] Besacier, L., Barnard, E., Karpov, A., and Schultz, T., Automatic speech recognition for under-resourced languages: A survey. *Speech Communication,* **2014**. 56: p. 85-100.

[16] Carki, K., Geutner, P., and Schultz, T. Turkish LVCSR: towards better speech recognition for agglutinative languages. in *Acoustics, Speech, and Signal Processing,* **2000***. ICASSP'00. Proceedings*.

[17] Kurimo, M., Puurula, A., Arisoy, E., Siivola, V., Hirsimäki, T., Pylkkönen, J., Alumäe, T., and Saraclar, M. Unlimited vocabulary speech recognition for agglutinative languages. in *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*. **2006**. Association for Computational Linguistics.

[18] Barras, C., Geoffrois, E., Wu, Z., and Liberman, M., Transcriber: development and use of a tool for assisting speech corpora production. Speech Communication, **2001**. 33(1): p. 5-22.

[19] Reddy, D. R., Speech recognition by machine: A review*. Proceedings of the IEEE,* 1976. 64(4): p. 501-531.

[20] Neto, J., Almeida, L., Hochberg, M., Martins, C., Nunes, L., Renals, S., and Robinson, T., Speaker-adaptation for hybrid HMM-ANN continuous speech recognition system*.* **1995**.

[21] Leggetter, C. J. and Woodland, P. C., Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models*. Computer Speech & Language,* **1995**. 9(2): p. 171-185.

[22] Xue, S., Jiang, H., Dai, L., and Liu, Q., Speaker adaptation of hybrid NN/HMM model for speech recognition based on singular value decomposition*. Journal of Signal Processing Systems*, **2016**. 82(2): p. 175-185.

[23] Huang, Z., Tang, J., Xue, S., and Dai, L. Speaker adaptation of RNN-BLSTM for speech recognition based on speaker code. in *Acoustics, Speech and Signal Processing (ICASSP),* **2016** *IEEE International Conference.*

[24] Gupta, V., Kenny, P., Ouellet, P., and Stafylakis, T. I-vector-based speaker adaptation of deep neural networks for french broadcast audio transcription. in *Acoustics, Speech and Signal Processing (ICASSP),* **2014** *IEEE International Conference on*. 2014. IEEE.

[25] Masmoudi, A., Khmekhem, M. E., Esteve, Y., Belguith, L. H., and Habash, N. A Corpus and Phonetic Dictionary for Tunisian Arabic Speech Recognition. in *LREC*. **2014**.

[26] Huang, X., Baker, J., and Reddy, R., A historical perspective of speech recognition*. Communications of the ACM*, 2014. 57(1): p. 94-103.

[27] Gong, Y., *Speech recognition in noisy environments: A survey.* Speech communication, **1995**. 16(3): p. 261-291.

[28] Akbacak, M., Burget, L., Wang, W., and Van Hout, J. *Rich system combination for keyword spotting in noisy and acoustically heterogeneous audio streams*. in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. **2013**.

[29] Fischer, A., Frinken, V., Bunke, H., and Suen, C. Y. *Improving hmm-based keyword spotting with character language models*. in *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*. **2013**.

[30] Smirnov, V., Ignatov, D., Gusev, M., Farkhadov, M., Rumyantseva, N., and Farkhadova, M., *A Russian Keyword Spotting System Based on Large Vocabulary Continuous Speech Recognition and Linguistic Knowledge.* Journal of Electrical and Computer Engineering, **2016**.

[31] Chen, G., Parada, C., and Heigold, G. *Small-footprint keyword spotting using deep neural networks*. in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. **2014**.

[32] Sainath, T. N. and Parada, C. *Convolutional neural networks for small-footprint keyword spotting*. in *Sixteenth Annual Conference of the International Speech Communication Association*. **2015**.

[33] Pathak, M. A. and Raj, B., *Privacy-preserving speaker verification and identification using gaussian mixture models.* IEEE Transactions on Audio, Speech, and Language Processing, **2013**. 21(2): p. 397-406.

[34] Li, M. and Liu, W. *Speaker verification and spoken language identification using a generalized i-vector framework with phonetic tokenizations and tandem features*. in *Fifteenth Annual Conference of the International Speech Communication Association*. **2014**.

[35] Variani, E., Lei, X., McDermott, E., Moreno, I. L., and Gonzalez-Dominguez, J. *Deep neural networks for small footprint text-dependent speaker verification*. in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. **2014**.

[36] Richardson, F., Reynolds, D., and Dehak, N., *Deep neural network approaches to speaker and language recognition.* IEEE Signal Processing Letters, **2015**. 22(10): p. 1671-1675.

[37] Zhang, Z., Wang, L., Kai, A., Yamada, T., Li, W., and Iwahashi, M., *Deep neural network-based bottleneck feature and denoising autoencoder-based dereverberation for distant-talking speaker identification.* EURASIP Journal on Audio, Speech, and Music Processing, **2015**

[38] Garcia-Romero, D., Snyder, D., Sell, G., Povey, D., and McCree, A. *Speaker diarization using deep neural network embeddings*. in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE*. **2017**.

[39] Larcher, A., Bonastre, J.-F., Fauve, B. G., Lee, K.-A., Lévy, C., Li, H., Mason, J. S., and Parfait, J.-Y. *ALIZE 3.0-open source toolkit for state-of-the-art speaker recognition*. in *Interspeech*. **2013**.

[40] Saedi, R., Lee, K. A., Kinnunen, T., Hasan, T., Fauve, B., Bousquet, P.-M., Khoury, E., Martinez, P. L. S., Kua, J. M. K., and You, C. *I4U submission to NIST SRE 2012: A large-scale collaborative effort for noise-robust speaker verification*. in *Interspeech*. **2013**.

[41] Snyder, D., Ghahremani, P., Povey, D., Garcia-Romero, D., Carmiel, Y., and Khudanpur, S. *Deep neural network-based speaker embeddings for end-to-*

end speaker verification. in *Spoken Language Technology Workshop (SLT), **2016** IEEE*.

[42] Lei, Y., Ferrer, L., McLaren, M., and Scheffer, N. *A deep neural network speaker verification system targeting microphone speech*. in *Fifteenth Annual Conference of the International Speech Communication Association*. **2014**.

[43] McLaren, M., Lei, Y., and Ferrer, L. *Advances in deep neural network approaches to speaker recognition*. in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. **2015**.

[44] Snyder, D., Garcia-Romero, D., Povey, D., and Khudanpur, S., *Deep Neural Network Embeddings for Text-Independent Speaker Verification.* Proc. Interspeech 2017, **2017**: p. 999-1003.

[45] Torfi, A., Nasrabadi, N. M., and Dawson, J., *Text-Independent Speaker Verification Using 3D Convolutional Neural Networks.* arXiv preprint arXiv:1705.09422, **2017**.

[46] Davis, S. B. and Mermelstein, P., *Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences*, in *Readings in speech recognition*. **1990**, Elsevier. p. 65-74.

[47] Hermansky, H., *Perceptual linear predictive (PLP) analysis of speech.* the Journal of the Acoustical Society of America, **1990**. 87(4): p. 1738-1752.

[48] Haigh, J. and Mason, J. *Robust voice activity detection using cepstral features*. in *TENCON'93. Proceedings. Computer, Communication, Control and Power Engineering. 1993 IEEE Region 10 Conference on*. **1993**.

[49] Hermansky, H., Ellis, D. P., and Sharma, S. *Tandem connectionist feature extraction for conventional HMM systems*. in *Acoustics, Speech, and Signal Processing, 2000. ICASSP'00. Proceedings. 2000 IEEE International Conference on*. **2000**.

[50] Sak, H., Senior, A., and Beaufays, F. *Long short-term memory recurrent neural network architectures for large scale acoustic modeling*. in *Fifteenth Annual Conference of the International Speech Communication Association*. **2014**.

[51] Dahl, G. E., Yu, D., Deng, L., and Acero, A., *Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition.* IEEE Transactions on audio, speech, and language processing, **2012**. 20(1): p. 30-42.

[52] Rabiner, L. R., *A tutorial on hidden Markov models and selected applications in speech recognition.* Proceedings of the IEEE, **1989**. 77(2): p. 257-286.

[53] Jaitly, N., Nguyen, P., Senior, A., and Vanhoucke, V. *Application of pretrained deep neural networks to large vocabulary speech recognition*. in *Thirteenth Annual Conference of the International Speech Communication Association*. **2012**.

[54] Deng, L., Li, J., Huang, J.-T., Yao, K., Yu, D., Seide, F., Seltzer, M., Zweig, G., He, X., and Williams, J. *Recent advances in deep learning for speech research at Microsoft*. in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. **2013**. IEEE.

[55] Senior, A., Heigold, G., Bacchiani, M., and Liao, H. *GMM-free DNN acoustic model training*. in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. **2014**.

[56] Baum, L. E. and Petrie, T., *Statistical inference for probabilistic functions of finite state Markov chains.* The annals of mathematical statistics, **1966**. 37(6): p. 1554-1563.

[57] Chan, W. and Lane, I. *Deep convolutional neural networks for acoustic modeling in low resource languages*. in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. **2015**.

[58] Kneser, R., Peters, J., and Klakow, D., *Speech recognition method with language model adaptation*. **2000**, Google Patents.

[59] Chen, S. F. and Goodman, J. *An empirical study of smoothing techniques for language modeling*. in *Proceedings of the 34th annual meeting on Association for Computational Linguistics*. **1996**. Association for Computational Linguistics.

[60] Stolcke, A., *Entropy-based pruning of backoff language models.* arXiv preprint cs/0006025, **2000**.

[61] Marge, M., Banerjee, S., and Rudnicky, A. I. *Using the Amazon Mechanical Turk for transcription of spoken language*. in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*. **2010**.

[62] Callison-Burch, C. and Dredze, M. *Creating speech and language data with Amazon's Mechanical Turk*. in *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*. **2010**. Association for Computational Linguistics.

[63] Fort, K., Adda, G., and Cohen, K. B., *Amazon mechanical turk: Gold mine or coal mine?* Computational Linguistics, **2011**. 37(2): p. 413-420.

[64] Sprouse, J., *A validation of Amazon Mechanical Turk for the collection of acceptability judgments in linguistic theory.* Behavior research methods, **2011**. 43(1): p. 155-167.

[65] Sorokin, A. and Forsyth, D. *Utility data annotation with amazon mechanical turk*. in *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08. IEEE Computer Society Conference on*. **2008**. IEEE.

[66] Ipeirotis, P. G., Provost, F., and Wang, J. *Quality management on amazon mechanical turk*. in *Proceedings of the ACM SIGKDD workshop on human computation*. **2010**. ACM.

[67] Buhrmester, M., Kwang, T., and Gosling, S. D., *Amazon's Mechanical Turk: A new source of inexpensive, yet high-quality, data?* Perspectives on psychological science, **2011**. 6(1): p. 3-5.

[68] Gauvain, J., Adda, G., Lamel, L., and Adda-Decker, M. *Transcribing broadcast news: The limsi nov96 hub4 system*. in *Proc. ARPA Speech Recognition Workshop*. **1997**.

[69] Jia, Y. and Culver, T. B., *Bootstrapped artificial neural networks for synthetic flow generation with a small data sample.* Journal of Hydrology, **2006**. 331(3): p. 580-590.

[70] Amodei, D., Ananthanarayanan, S., Anubhai, R., Bai, J., Battenberg, E., Case, C., Casper, J., Catanzaro, B., Cheng, Q., and Chen, G. *Deep speech 2: End-to-end speech recognition in english and mandarin*. in *International Conference on Machine Learning*. **2016**.

[71] Hannun, A., Case, C., Casper, J., Catanzaro, B., Diamos, G., Elsen, E., Prenger, R., Satheesh, S., Sengupta, S., and Coates, A., *Deep speech: Scaling up end-to-end speech recognition.* arXiv preprint arXiv:1412.5567, **2014**.

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

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

[74] Cybenko, G., *Approximation by superpositions of a sigmoidal function.* Mathematics of Control, Signals, and Systems (MCSS), **1989**. 2(4): p. 303-314.

[75] Bridle, J. S., *Alpha-nets: a recurrent 'neural'network architecture with a hidden Markov model interpretation.* Speech Communication, **1990**. 9(1): p. 83-92.

[76] Deng, L.-Y., *The cross-entropy method: a unified approach to combinatorial optimization, Monte-Carlo simulation, and machine learning*. **2006**, Taylor & Francis.

[77] Rumelhart, D. E., Hinton, G. E., and Williams, R. J., *Learning internal representations by error propagation*. **1985**, California Univ San Diego La Jolla Inst for Cognitive Science.

[78] Bottou, L., *Stochastic gradient learning in neural networks.* Proceedings of Neuro-Nımes, **1991**. 91(8).

[79] Bourlard, H. A. and Morgan, N., *Connectionist speech recognition: a hybrid approach*. Vol. 247. **2012**: Springer Science & Business Media.

[80] Dahl, G. E., Sainath, T. N., and Hinton, G. E. *Improving deep neural networks for LVCSR using rectified linear units and dropout*. in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. **2013**.

[81] Bengio, Y., Lamblin, P., Popovici, D., and Larochelle, H. *Greedy layer-wise training of deep networks*. in *Advances in neural information processing systems*. **2007**.

[82] Abdel-Hamid, O., Mohamed, A.-r., Jiang, H., and Penn, G. *Applying convolutional neural networks concepts to hybrid NN-HMM model for speech recognition*. in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*. **2012**.

[83] Chan, W., Jaitly, N., Le, Q., and Vinyals, O. *Listen, attend and spell: A neural network for large vocabulary conversational speech recognition*. in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. **2016**.

[84] Collobert, R., Puhrsch, C., and Synnaeve, G., *Wav2Letter: an End-to-End ConvNet-based Speech Recognition System.* CoRR, **2016**. abs/1609.03193.

[85] Baker, J. M., Deng, L., Glass, J., Khudanpur, S., Lee, C.-H., Morgan, N., and O'Shaughnessy, D., *Developments and directions in speech recognition and understanding, Part 1 [DSP Education].* IEEE Signal Processing Magazine, **2009**. 26(3).

[86] Furui, S., *Digital speech processing: synthesis, and recognition*. **2000**: CRC Press.

[87] Orr, G. B. and Müller, K.-R., *Neural networks: tricks of the trade*. **2003**: Springer.

[88] Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. R., *Improving neural networks by preventing co-adaptation of feature detectors.* arXiv preprint arXiv:1207.0580, **2012**.

[89] Kim, Y., Jernite, Y., Sontag, D., and Rush, A. M. *Character-Aware Neural Language Models*. in *AAAI*. **2016**.

[90] Chen, X., Liu, X., Gales, M. J., and Woodland, P. C. *Recurrent neural network language model training with noise contrastive estimation for speech recognition*. in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. **2015**.

[91] Yao, K., Zweig, G., Hwang, M.-Y., Shi, Y., and Yu, D. *Recurrent neural networks for language understanding*. in *Interspeech*. **2013**.

[92] Bahdanau, D., Cho, K., and Bengio, Y., *Neural machine translation by jointly learning to align and translate.* arXiv preprint arXiv:1409.0473, **2014**.

[93] Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y., *Learning phrase representations using RNN encoder-decoder for statistical machine translation.* arXiv preprint arXiv:1406.1078, **2014**.

[94] Devlin, J., Zbib, R., Huang, Z., Lamar, T., Schwartz, R. M., and Makhoul, J. *Fast and Robust Neural Network Joint Models for Statistical Machine Translation*. in *ACL (1)*. **2014**.

[95] Cho, K., Van Merriënboer, B., Bahdanau, D., and Bengio, Y., *On the properties of neural machine translation: Encoder-decoder approaches.* arXiv preprint arXiv:1409.1259, **2014**.

[96] Gregor, K., Danihelka, I., Graves, A., Rezende, D. J., and Wierstra, D., *DRAW: A recurrent neural network for image generation.* arXiv preprint arXiv:1502.04623, **2015**.

[97] Mao, J., Xu, W., Yang, Y., Wang, J., and Yuille, A. L., *Explain images with multimodal recurrent neural networks.* arXiv preprint arXiv:1410.1090, **2014**.

[98] Vinyals, O., Toshev, A., Bengio, S., and Erhan, D. *Show and tell: A neural image caption generator*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. **2015**.

[99] Mao, J., Xu, W., Yang, Y., Wang, J., Huang, Z., and Yuille, A., *Deep captioning with multimodal recurrent neural networks (m-rnn).* arXiv preprint arXiv:1412.6632, **2014**.

[100] Oord, A. v. d., Kalchbrenner, N., and Kavukcuoglu, K., *Pixel recurrent neural networks.* arXiv preprint arXiv:1601.06759, **2016**.

[101] Graves, A., Mohamed, A.-r., and Hinton, G. *Speech recognition with deep recurrent neural networks*. in *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*. **2013**. IEEE.

[102] Sak, H., Senior, A., Rao, K., and Beaufays, F., *Fast and accurate recurrent neural network acoustic models for speech recognition.* arXiv preprint arXiv:1507.06947, **2015**.

[103] Hochreiter, S. and Schmidhuber, J., *Long short-term memory.* Neural computation, **1997**. 9(8): p. 1735-1780.

[104] LeCun, Y., Bengio, Y., and Hinton, G., *Deep learning.* Nature, **2015**. 521(7553): p. 436-444.

[105] Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R., and Schmidhuber, J., *LSTM: A search space odyssey.* IEEE transactions on neural networks and learning systems, **2017**.

[106] Sutskever, I., Vinyals, O., and Le, Q. V. *Sequence to sequence learning with neural networks*. in *Advances in neural information processing systems*. **2014**.

[107] Olah, C., *Understanding lstm networks.* GITHUB blog, posted on August, 2015. 27: p. **2015**.

[108] Kingma, D. P. and Ba, J., *Adam: A Method for Stochastic Optimization.* CoRR, **2014**. abs/1412.6980.

[109] Andrychowicz, M., Denil, M., Colmenarejo, S. G., Hoffman, M. W., Pfau, D., Schaul, T., and de Freitas, N., *Learning to learn by gradient descent by gradient descent.* CoRR, **2016**. abs/1606.04474.

[110] Mnih, V., Heess, N., Graves, A., and Kavukcuoglu, K., *Recurrent Models of Visual Attention.* CoRR, **2014**. abs/1406.6247.

[111] Chan, W., Jaitly, N., Le, Q. V., and Vinyals, O., *Listen, Attend and Spell.* CoRR, **2015**. abs/1508.01211.

[112] Luong, M.-T., Pham, H., and Manning, C. D., *Effective Approaches to Attention-based Neural Machine Translation.* CoRR, **2015**. abs/1508.04025.

[113] Graves, A., Fernández, S., Gomez, F., and Schmidhuber, J. *Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks*. in *Proceedings of the 23rd international conference on Machine learning*. **2006**. ACM.

[114] Miao, Y., Gowayyed, M., and Metze, F. *EESEN: End-to-end speech recognition using deep RNN models and WFST-based decoding*. in *Automatic Speech Recognition and Understanding (ASRU), 2015 IEEE Workshop on*. **2015**.

[115] Li, J., Zhang, H., Cai, X., and Xu, B. *Towards end-to-end speech recognition for chinese mandarin using long short-term memory recurrent neural networks*. in *Sixteenth Annual Conference of the International Speech Communication Association*. **2015**.

[116] Chorowski, J., Bahdanau, D., Cho, K., and Bengio, Y., *End-to-end continuous speech recognition using attention-based recurrent NN: first results.* arXiv preprint arXiv:1412.1602, **2014**.

[117] Chorowski, J. K., Bahdanau, D., Serdyuk, D., Cho, K., and Bengio, Y. *Attention-based models for speech recognition*. in *Advances in Neural Information Processing Systems*. **2015**.

[118] El Hihi, S. and Bengio, Y. *Hierarchical recurrent neural networks for long-term dependencies*. in *Advances in neural information processing systems*. **1996**.

[119] Sainath, T. N., Vinyals, O., Senior, A., and Sak, H. *Convolutional, long short-term memory, fully connected deep neural networks*. in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. **2015**.

[120] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., and Devin, M., *Tensorflow: Large-scale machine learning on heterogeneous distributed systems.* arXiv preprint arXiv:1603.04467, **2016**.

[121] Povey, D., Kanevsky, D., Kingsbury, B., Ramabhadran, B., Saon, G., and Visweswariah, K. *Boosted MMI for model and feature-space discriminative training*. in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*. **2008**.

[122] Ko, T., Peddinti, V., Povey, D., and Khudanpur, S. *Audio augmentation for speech recognition*. in *INTERSPEECH*. **2015**.

[123] Povey, D., Peddinti, V., Galvez, D., Ghahremani, P., Manohar, V., Na, X., Wang, Y., and Khudanpur, S. *Purely Sequence-Trained Neural Networks for ASR Based on Lattice-Free MMI*. in *INTERSPEECH*. **2016**.

[124] Luong, M.-T., Sutskever, I., Le, Q. V., Vinyals, O., and Zaremba, W., *Addressing the rare word problem in neural machine translation.* arXiv preprint arXiv:1410.8206, **2014**.

[125] Jean, S., Cho, K., Memisevic, R., and Bengio, Y., *On using very large target vocabulary for neural machine translation.* arXiv preprint arXiv:1412.2007, **2014**.

[126] Pennington, J., Socher, R., and Manning, C. *Glove: Global vectors for word representation*. in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. **2014**.

[127] Stolcke, A. *SRILM-an extensible language modeling toolkit*. in *Interspeech*. **2002**.

[128] Bisani, M. and Ney, H. *Investigations on joint-multigram models for grapheme-to-phoneme conversion*. in *Seventh International Conference on Spoken Language Processing*. **2002**.

[129] Pan, S. J. and Yang, Q., *A Survey on Transfer Learning.* IEEE Trans. on Knowl. and Data Eng., **2010**. 22(10): p. 1345-1359.

[130] Huang, J.-T., Li, J., Yu, D., Deng, L., and Gong, Y. *Cross-language knowledge transfer using multilingual deep neural network with shared*

hidden layers. in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. **2013**. IEEE.

[131] Kunze, J., Kirsch, L., Kurenkov, I., Krug, A., Johannsmeier, J., and Stober, S., *Transfer Learning for Speech Recognition on a Budget.* arXiv preprint arXiv:1706.00290, **2017**.

[132] Caruana, R., *Multitask learning*, in *Learning to learn*. **1998**, Springer. p. 95-133.

[133] Heigold, G., Vanhoucke, V., Senior, A., Nguyen, P., Ranzato, M. A., Devin, M., and Dean, J. *Multilingual acoustic models using distributed deep neural networks*. in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. **2013**. IEEE.

[134] Ghahremani, P., Manohar, V., Hadian, H., Povey, D., and Khudanpur, S., *INVESTIGATION OF TRANSFER LEARNING FOR ASR USING LF-MMI TRAINED NEURAL NETWORKS.*

[135] Panayotov, V., Chen, G., Povey, D., and Khudanpur, S. *Librispeech: an ASR corpus based on public domain audio books*. in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. **2015**.

[136] Bahdanau, D., Chorowski, J., Serdyuk, D., Brakel, P., and Bengio, Y. *End-to-end attention-based large vocabulary speech recognition*. in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. **2016**.

[137] Yao, K., Yu, D., Seide, F., Su, H., Deng, L., and Gong, Y. *Adaptation of context-dependent deep neural networks for automatic speech recognition*. in *Spoken Language Technology Workshop (SLT), 2012 IEEE*. **2012**.

[138] Goldberg, Y., *A Primer on Neural Network Models for Natural Language Processing.* J. Artif. Intell. Res.(JAIR), **2016**. 57: p. 345-420.

[139] Cho, E., Niehues, J., Kilgour, K., and Waibel, A. *Punctuation insertion for real-time spoken language translation*. in *Proceedings of the Eleventh International Workshop on Spoken Language Translation*. **2015**.

[140] Gravano, A., Jansche, M., and Bacchiani, M. *Restoring punctuation and capitalization in transcribed speech*. in *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*. **2009**. IEEE.

[141] Lu, W. and Ng, H. T. *Better punctuation prediction with dynamic conditional random fields*. in *Proceedings of the 2010 conference on empirical methods in natural language processing*. **2010**. Association for Computational Linguistics.

[142] Zhang, D., Wu, S., Yang, N., and Li, M. *Punctuation Prediction with Transition-based Parsing*. in *ACL (1)*. **2013**.

[143] Peitz, S., Freitag, M., Mauser, A., and Ney, H. *Modeling punctuation prediction as machine translation*. in *International Workshop on Spoken Language Translation (IWSLT)* **2011**.

[144] Kolář, J., Shriberg, E., and Liu, Y. *Using prosody for automatic sentence segmentation of multi-party meetings*. in *International Conference on Text, Speech and Dialogue*. **2006**. Springer.

[145] Kolář, J. and Lamel, L. *Development and evaluation of automatic punctuation for French and English speech-to-text*. in *Thirteenth Annual Conference of the International Speech Communication Association*. **2012**.

[146] Mikolov, T., Chen, K., Corrado, G., and Dean, J., *Efficient Estimation of Word Representations in Vector Space.* CoRR, **2013**. abs/1301.3781.

ÖZGEÇMİŞ

Kimlik Bilgileri

Adı Soyadı	: Behnam ASEFISARAY
Doğum Yeri	: İran
Medeni Hali	: Evli
E-posta	: bh.asefi@gmail.com
Adresi	: Bilkent CyberPark SESTEK Firması, 06800, Çankaya, Ankara, Türkiye

Eğitim

Lisans	: Azad University, İran (2005-2009)
Yüksek Lisans	: -
Bütünleşik Doktora	: Hacettepe Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Bölümü, Ankara, Türkiye

Yabancı Dil ve Düzeyi

İngilizce	: ÜDS – 76.25 (2010)

İş Deneyimi

Kıdemli Ar-Ge Uzmanı	: SESTEK - Ses, İletişim, Bilgisayar ve Çağrı Merkezi Çözümleri – (2015 - Halen)
Kıdemli Yazılım Mühendisi	: MANTİS Yazılım ve Danışmanlık (2013-2015)
Yazılım Mühendisi	: MANTİS Yazılım ve Danışmanlık (2010-2013)

Deneyim Alanları

Otomatik Konuşma Tanıma, Akustik Modelleme, Dil Modelleme, Makine Öğrenimi, Derin Öğrenme, Sinyal İşleme

Tezden Üretilmiş Projeler ve Bütçeleri

00815.STZ.2011-1 no'lu SanTez Projesi: Mobil Sistemlerde Ses Tanıma ile Türkçe-İngilizce Tercüme Sistemi . Bütçesi: 275.685,00 TL – 2011-2014

Tezden Üretilmiş Yayınlar

- Asefisaray, B., E. Mengüşoğlu, M. Hacıömeroğlu, and H. Sever, How Does Language Model Training Data Size Effects Speech Recognition Accuracy For Turkish Language - Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi, 2016. 22(2): p. 100-105.
- Asefisaray, B., E. Mengüşoğlu and H. Sever, Compilation of a Transcribed Speech Corpus for Turkish LVCSR from Movies, International Journal of Speech Technology (Submissioned)

Tezden Üretilmiş Tebliğ ve/veya Poster Sunumu ile Katıldığı Toplantılar

Asefisaray, B. and H. Sever The Use of Automatic Speaker Recognition in Forensic Science, PolTek Workshop Istanbul 2013

HACETTEPE ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
YÜKSEK LİSANS/DOKTORA TEZ ÇALIŞMASI ORJİNALLİK RAPORU

HACETTEPE ÜNİVERSİTESİ
FEN BİLİMLER ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI BAŞKANLIĞI'NA

Tarih: 15/01/2018

Tez Başlığı / Konusu: Uçtan-Uca Konuşma Tanıma Modeli: Turkçe'deki Deneyler

Yukarıda başlığı/konusu gosterilen tez çalışmamın a) Kapak sayfası, b) Giriş, c) Ana bolumler d) Sonuç kısımlarından oluşan toplam 125 sayfalık kısmına ilişkin, 15/01/2018 tarihinde şahsım/tez danışmanım tarafından *Turnıtın* adlı intihal tespit programından aşağıda belirtilen filtrelemeler uygulanarak alınmış olan orijinallik raporuna gore, tezimin benzerlik oranı % 2 'dir.

Uygulanan filtrelemeler:

1- Kaynakça hariç
2- Alıntılar hariç
3- 5 kelimeden daha az ortüşme içeren metin kısımları hariç

Hacettepe Üniversitesi Fen Bilimleri Enstitüsü Tez Çalışması Orjinallik Raporu Alınması ve Kullanılması Uygulama Esasları'nı inceledim ve bu Uygulama Esasları'nda belirtilen azami benzerlik oranlarına gore tez çalışmamın herhangi bir intihal içermediğini; aksinin tespit edileceği muhtemel durumda doğabilecek her turlu hukuki sorumluluğu kabul ettiğimi ve yukarıda vermiş olduğum bilgilerin doğru olduğunu beyan ederim.

Gereğını saygılarımla arz ederim.

15.01.2018

Adı Soyadı: BEHNAM ASEFISARAY
Öğrenci No: N10164995
Anabilim Dalı: BİLGİSAYAR MUHENDİSLİĞİ
Programı: BUTUNLEŞİK DOKTORA
Statüsü: ☐ Y.Lisans ☐ Doktora ☒ Butunleşik Dr.

DANIŞMAN ONAYI

UYGUNDUR.

Prof.Dr. Hayri SEVER