

**TÜRKÇE HİS ANALİZİNDE OPTİMİZASYON VE ÖN-
EĞİTİMLİ MODELLERİN KULLANIMI**

**USE OF OPTIMIZATION AND PRE-TRAINED MODELS
IN TURKISH EMOTION ANALYSIS**

Alaettin UÇAN

PROF. DR. Ebru SEZER

Tez Danışmanı

Hacettepe Üniversitesi

Lisansüstü Eğitim-Öğretim ve Sınav Yönetmeliğinin
Bilgisayar Mühendisliği Anabilim Dalı için Öngördüğü

DOKTORA TEZİ olarak hazırlanmıştır.

2020

Eşim ve Kızlarıma

ÖZET

TÜRKÇE HİS ANALİZİNDE OPTİMİZASYON VE ÖN-EĞİTİMLİ MODELLERİN KULLANIMI

Alaettin UÇAN

Doktora, Bilgisayar Mühendisliği Anabilim Dalı

Tez Danışmanı: Prof. Dr. Ebru SEZER

Aralık 2020, 127 sayfa

His analizi sayısal içerikte yer alan hislerin otomatik olarak belirlenmesi işlemidir. İnternet ve sayısal iletişimde yaşanan baş döndürücü teknolojik gelişmeler ve bunların etkisiyle oluşan toplumsal değişimler sonucunda ortaya çıkan büyük verinin his analizi yöntemleriyle damıtılarak kıymetlendirilmesi bir gereklilik haline gelmiştir. Ancak makine öğrenmesi yöntemleriyle his analizi yapmak için dengeli sınıf dağılımına sahip çok miktarda etiketli veriye ihtiyaç duyulmaktadır. Öte yandan hislerin jest ve mimikler olmadan sadece kelimelerle ifade edilmesi probleme büyük bir belirsizlik katmaktadır.

Bu tez kapsamında, makine öğrenmesi yöntemleriyle yüksek performanslı Türkçe his analizi yapmak amaçlanmıştır. Bu amaçla belirsizliği arttıran dengesiz sınıf dağılımı ve veri azlığı problemlerine çözümler önerilerek his analizinin performansı arttırılmaktadır. Çalışmada dengesiz dağılım gösteren veri kümelerinde yer alan tüm sınıfların başarısını

ölçmek için sınıf içi performanslar kullanılmaktadır. Belirsizliği azaltmak amacıyla geliştirilen optimum his vektörleri yöntemiyle derin öğrenme karar mekanizmasına getirilen yenilik sayesinde elde edilen genel doğruluk performansları korunurken, sınıf içi his analizi performansları arttırılmıştır.

Ayrıca Türkçe için ilk kez, ön-eğitilmiş dil modelleri his analizine adapte edilerek, çok miktarda etiketli veri ihtiyacı olmayan ön-eğitilmiş his modeli geliştirilmiştir. Geliştirilen yöntemden elde edilen sonuçlar farklı veri kümeleri ve öğrenme yöntemleriyle karşılaştırmalı olarak sunulmaktadır. Ön-eğitilmiş his modeliyle Türkçe his veri kümeleri üzerinde gerçekleştirilen deneyler sonucunda alandaki en yüksek performans elde edilmiştir.

Önerilen çalışma Türkçe kaynaklarda daha isabetli kamuoyu yoklaması, müşteri ilişkileri yönetimi, marka yönetimi, siber zorbalığın tespiti, seçimlerdeki yönelimin tespiti, partizan yorumları belirleme gibi üst seviye yapay zekâ görevlerin daha başarılı olarak yapılmasına olanak sağlayacaktır. Ayrıca bu alandaki araştırmacılar için bir kaynak olacak ve Türkçe metin madenciliği görevlerinde kullanılabilir.

Anahtar Kelimeler: Türkçe His Analizi, Derin Öğrenme, Ön-eğitilmiş Dil Modeli, Optimizasyon

ABSTRACT

USE OF OPTIMIZATION AND PRE-TRAINED MODELS IN TURKISH EMOTION ANALYSIS

Alaettin UÇAN

Doctor of Philosophy, Department of Computer Engineering

Supervisor: Prof. Dr. Ebru SEZER

December 2020, 127 pages

Emotion analysis means to automatic determination of the emotions in any digital content. It has become a necessity to distill and evaluate with emotion analysis the big data that emerged as a result of the dizzying technological developments in the internet and digital communication and the social changes caused by these. However, a large amount of labeled data with proportional class distribution is needed to perform emotion analysis with machine learning methods. On the other hand, expressing feelings only in words without gestures adds a great uncertainty to the problem.

Herein, it is aimed to perform high performance Turkish emotion analysis with machine learning techniques. For this purpose, the performance of emotion analysis is increased by proposing solutions to unbalanced class distribution which increases uncertainty, and data scarcity. In the study, inner class performances are used to measure the success of

all classes in datasets showing imbalanced distribution. While maintaining the general accuracy performances achieved by the innovation brought to the deep learning decision mechanism with the optimum emotion vectors method developed to reduce uncertainty, the inner class emotion analysis performances were increased.

In addition, for the first time in Turkish, pre-trained language models were adapted to emotion analysis, and a pre-trained emotion model, which does not require large amounts of labeled data, was developed. The results obtained from the developed method are presented comparatively with different data sets and learning methods. As a result of the experiments performed on the Turkish emotion data sets with the pre-trained emotion model, the highest performance in the field was obtained.

The proposed study will enable high-level artificial intelligence tasks such as more accurate public opinion polling, customer relationship management, brand management, detection of cyber bullying, determination of the tendency in elections, determining partisan comments in Turkish resources. It will also be a resource for researchers in this field, and can be used in Turkish text mining tasks.

Keywords: Turkish Emotion Analysis, Deep Learning, Pre-trained Language Model, Optimization

TEŐEKKÜR

Lisansüstü eğitimimin her aşamasında yanımda olan, bilgeliđi ve hoşgörüsüyle ufkumu açan ve yardımını hiçbir zaman esirgemeyen kıymetli danışmanım Prof. Dr. Ebru SEZER'e;

Tez izleme komiteleri boyunca yapıcı fikir ve önerileriyle tezime yaptıkları katkılardan dolayı Prof. Dr. Ahmet COŐAR ve Prof. Dr. Őükrü Haluk AKALIN'a;

Kıymetli vakitlerini ayırdıkları ve katkılarından dolayı, tez savunma jürimde bulunan Dr. Öğr. Üyesi Fuat AKAL ve Dr. Öğr. Üyesi Öner BARUT'a;

Tez çalışmam boyunca verdiği motivasyon ve akademik desteklerinden dolayı Dr. Murat DÖRTERLER'e;

Biricik kızlarım Asel Zeynep ve Asya Lina'ya;

Gösterdiği sabır ve desteğinden dolayı değerli eşim Zeynep UŐAN'a;

Tüm içtenliğimle teşekkür ederim.

İÇİNDEKİLER

ÖZET.....	i
ABSTRACT	iii
TEŞEKKÜR	v
İÇİNDEKİLER.....	vi
ŞEKİLLER DİZİNİ.....	ix
ÇİZELGELER DİZİNİ	x
SİMGELER VE KISALTMALAR.....	xi
1. GİRİŞ	1
1.1. Problemin Tanımı.....	1
1.2. Amaç ve Motivasyon	5
1.3. Özgün Değer	6
1.4. Tez Organizasyonu.....	6
2. ALAN BİLGİSİ VE ALAN YAZIN ÖZETİ	8
2.1. His Modelleri.....	8
2.1.1. Ekman'ın Temel Hisler Teorisi.....	8
2.1.2. Plutchik'in Hislerin Çarkı Teorisi	9
2.1.3. Russel'in Circumplex His Modeli.....	10
2.2. His Sınıflama.....	11
2.2.1. Anahtar Kelime Tabanlı Sınıflama	12
2.2.2. Sözlük Tabanlı Sınıflama	12
2.2.3. Geleneksel Makine Öğrenmesi Yöntemleri	13
2.2.4. Derin Öğrenme Yöntemleri.....	16
2.2.4.1. Evrişimli Sinir Ağları (CNN).....	18
2.2.4.2. Tekrarlayan Sinir Ağları (RNN)	21
2.2.4.2.1. Uzun Kısa-Sürelili Bellek Ağları (LSTM).....	22
2.2.4.2.2. Geçitli Tekrarlayan Birim Ağları (GRU).....	24

2.2.5. Ön Eğitimli Modeller ile His Sınıflama.....	25
2.2.5.1. Ön Eğitimli Kelime Özyerleşikleri	27
2.2.5.1.1. Word2Vec Kelime Özyerleşigi.....	27
2.2.5.1.2. GloVe Kelime Özyerleşigi.....	29
2.2.5.1.3. FastText Kelime Özyerleşigi	30
2.2.5.2. Ön Eğitimli Dil Modelleri.....	31
2.2.5.2.1. BERT Dil Modeli.....	31
2.2.5.2.2. Diğer Dil Modelleri	33
2.3. Optimizasyon Algoritmalarıyla His Sınıflama	34
2.3.1. Parçacık Sürü Optimizasyonu (PSO).....	34
2.3.1.1. Gri Kurt Optimizasyonu (GWO)	35
2.4. Türkçe His Sınıflama Çalışmaları.....	37
2.5. His Analizi Uygulamaları	38
2.6. His Analizi Çalışmalarının Tarihsel Gelişimi.....	39
3. OPTIMUM HİS VEKTÖRLERİ VE DERİN ÖĞRENME İLE HİS ANALİZİ.....	42
3.1.1. Yöntem.....	44
3.1.1.1. Optimum His Vektörleri Oluşturma	46
3.1.1.2. Derin Öğrenme Modeli Uygulama Ayrıntıları	49
3.1.2. His Veri Kümeleri.....	49
3.1.3. Performans Metrikleri.....	52
3.1.4. Deney Sonuçları.....	53
4. ÖN EĞİTİMLİ DİL MODELLERİ İLE TÜRKÇE HİS ANALİZİ	61
4.1.1. Yöntem.....	63
4.1.2. His Veri Kümeleri.....	65
4.1.3. Deneysel Çalışmalar	66
4.1.4. Performans metrikleri	67
4.1.5. Deney Sonuçları.....	67
4.1.6. Ön Eğitimli His Modeli için Optimum His Vektörleri	72
5. SONUÇLAR VE GELECEK ÇALIŞMALAR	75
5.1. Sonuçlar	75
5.2. Gelecek Çalışmalar	76

5. KAYNAKLAR.....	78
EKLER	98
EK 1 - Tezden Türetilmiş Yayınlar	98
EK 2 - Tezden Türetilmiş Bildiriler	99
EK 3 - Tez Çalışması Orjinallik Raporu	100
ÖZGEÇMİŞ	101

ŞEKİLLER DİZİNİ

Şekil 2.1 Ekman Temel Hislerine Ait Örnek Yüz İfadeleri.....	9
Şekil 2.2 Plutchik'in His Çarkı modeli	10
Şekil 2.3 Russel'in Circumplex His Modeli	11
Şekil 2.4 Doğal ve Yapay Sinir Ağları [61].....	17
Şekil 2.5 Makine öğrenmesi ve derin öğrenme farkları.....	18
Şekil 2.6 Evrişim (Convolution) işlemi [69]	19
Şekil 2.7 CNN ile metin sınıflama yöntemi [73]	20
Şekil 2.8 RNN nöronunun açık hali [81]	22
Şekil 2.9 LSTM hücre yapısı [71]	23
Şekil 2.10 Geçitli Tekrarlayan Birim yapısı [94]	24
Şekil 2.11 Word2Vec algoritması [106]	28
Şekil 2.12 Word2Vec anlam uzayında kelime vektörleri	28
Şekil 2.13 BERT girdi gösterimi [9].....	32
Şekil 2.14 Parçacık sürü optimizasyon örneği	35
Şekil 2.15 Gri kurt sürüsü hiyerarşisi [183].....	36
Şekil 2.16 Gri kurt algoritması sözde kodu [183].....	37
Şekil 3.17 Önerilen optimum his vektörleri ve derin öğrenme ile his analizi yöntemi ..	45
Şekil 4.18 Önerilen ön eğitilmiş dil modelleri ile Türkçe his analizi yöntemi	64
Şekil 4.19 His sınıflama sonuçları	71

ÇİZELGELER DİZİNİ

Çizelge 2.1 Metin üzerinde gövdeleme ve parçalama.....	14
Çizelge 2.2 Kelime çantası ve kelime n-gramları	14
Çizelge 2.3 Alan yazın çalışmalarının tarihsel özeti	40
Çizelge 3.4 İngilizce his veri kümeleri istatistikleri.....	50
Çizelge 3.5 Türkçe his veri kümeleri istatistikleri	51
Çizelge 3.6 Yaklaşımların genel doğruluk değerleriyle karşılaştırılması	54
Çizelge 3.7 Türkçe Optimum his vektörleri ve ArgMax yöntemi karşılaştırması	54
Çizelge 3.8 İngilizce Optimum his vektörleri ve ArgMax yöntemi karşılaştırması	55
Çizelge 3.9 Optimum His Vektörleri ve ArgMax ortalama metriklerle karşılaştırma....	56
Çizelge 3.10 CrowdFlower veri kümesi POEV ve ArgMax Anma (R) karşılaştırması .57	
Çizelge 3.11 TURTED veri kümesi POEV ve ArgMax doğruluk tablosu	57
Çizelge 3.12 POEV ve Bostan yaklaşımları performans karşılaştırması	58
Çizelge 3.13 POEV ve ArgMax Belirsizlik giderimi karşılaştırması	59
Çizelge 4.14 BERTurk[12] ön-eğitilmiş dil modeliyle Türkçe His Sınıflama Sonuçları.67	
Çizelge 4.15 TREMO Test kümesi karmaşıklık matrisi	68
Çizelge 4.16 TURTED Test kümesi karmaşıklık matrisi	68
Çizelge 4.17 Farklı ön-eğitilmiş dil modelleriyle Türkçe his analizi sonuçları.....	69
Çizelge 4.18 Kelime Özyerleşiklerinin derin öğrenme sonuçlarına etkisi.....	69
Çizelge 4.19 TREMO Önerilen yöntem ve diğer yöntemlerin sonuçları.....	70
Çizelge 4.20 TURTED Önerilen yöntem ve diğer yöntemlerin sonuçları.....	70
Çizelge 4.21 Gerçekleştirilen tüm deney sonuçları.....	72
Çizelge 4.22 Ön Eğitilmiş His Modeli için Optimum His Vektörleri etkisi.....	73
Çizelge 4.23 TREMO Ön Eğitilmiş His Modeli ArgMax ve NOEV doğruluk matrisi....	74
Çizelge 4.24 TURTED Ön Eğitilmiş His Modeli ArgMax ve POEV doğruluk matrisi ..	74

SİMGELER VE KISALTMALAR

Simgeler

FN	Yanlış Negatifler
FP	Yanlış Pozitifler
P	Precision (Kesinlik)
R	Recall (Anma)
TN	Doğru Negatifler
TP	Doğru Pozitifler

Kısaltmalar

Bi-GRU	Bidirectional GRU
Bi-LSTM	Bidirectional LSTM
BOW	Bag of Words
CBOW	Continuous Bag Of Words
CNN	Convolutional Neural Networks
DDİ	Doğal Dil İşleme
GRU	Gated Recurrent Unit
GWO	Grey Wolf Optimization
KNN	K Nearest Neighbors
LSTM	Long-Short Term Memory
MLP	Multi-Layer Perceptron
OOV	Out Of Vocabulary
PSO	Particle Swarm Optimization
RNN	Recurrent Neural Networks
SVM	Support Vector Machines
YSA	Yapay Sinir Ağları

1. GİRİŞ

1.1. Problemin Tanımı

Günümüzde teknolojik gelişmeler sayesinde ve salgın gibi toplum sağlığını etkileyen olumsuz gelişmeler sebebiyle insan-bilgisayar etkileşiminin limitleri zorlanmaktadır. Eskiden yüz yüze yapılan toplantı, alışveriş, bankacılık, emlakçılık, eğitim hatta doktor muayenesi gibi etkileşimler bile bilgisayar sistemleri yardımıyla uzaktan yapılabilmektedir. İnsanlarla sesli, yazılı ve görüntülü etkileşim içinde olan bilgisayar sistemlerinin daha insansı bir etki uyandırabilmesi için anlamsal boyuttan öte duygusal olarak da tatmin edici bir iletişim kurması bir gereklilik haline gelmiştir. Çünkü insanlar arasındaki iletişim salt metinlerden oluşmamaktadır. Günlük hayatta iletişime duygu ve hisler eşlik etmekte ve anlam katmaktadır. Örneğin kitap, film ya da resim gibi sanat eserleri ancak güçlü hisler yansıttıklarında ilgi çekici ve anlamlı olmaktadır. Bu sebeple insan-bilgisayar etkileşiminde hislerin sayısallaştırılarak iletişimin bir parçası haline getirilmesi gerekmektedir.

İnsan-bilgisayar etkileşiminin yanı sıra bilgisayar üzerinden gerçekleşen insanlar arası iletişimde de duygu ve hisler önemli yer tutmaktadır. İnsanlar bir konu, ürün, film, restoran ya da otel hakkındaki fikirlerini, duygularını ve tecrübelerini internet üzerinden paylaşmaktadır. Bu paylaşımlar okuyucuda konu hakkında bir ön fikir oluşmasını sağlamakta ve nasıl bir deneyim elde edeceğini kestirmesine yardımcı olmaktadır. Örneğin; izleyiciler genelde filmi daha önce izlemiş olanların film hakkında yaptıkları yorumlara bakarak film seçimi yapmaktadır. Aynı şekilde tatil yapmak isteyen bir müşteri oteller hakkında paylaşılan tecrübeleri inceleyerek tesis seçimi yapmaktadır. Öte yandan hizmet sağlayıcılar için müşteriler tarafından ürünleri hakkında yapılan yorumlar oldukça kıymetlidir. Bu sebeple insanlar tarafından yapılan bu paylaşımların bilgisayar yardımıyla otomatik olarak anlamlı bilgilere dönüştürülmesi gerekmektedir.

İnternet ve sosyal medya kullanımının artmasıyla değerlendirilmesi gereken yorum sayısı binlerce olabilmektedir. Bu gibi durumlarda insan algısıyla tüm yazılanları okuyup değerlendirmek neredeyse imkânsızdır. Bu problemin çözümü, konu hakkında yapılan

yorumları bilgisayar yardımıyla otomatik olarak değerlendirmektir. Bu yönüyle his analizi için metin sınıflama problemine benzer çözümler önerilmiştir. Önce cümleleri olumlu, olumsuz veya tarafsız şeklinde sınıflayan duygu analizi çalışmaları yapılmıştır. Devamında ise cümle içerisindeki öfke, korku, sevinç, üzüntü vb. temel hisleri belirlemeyi amaçlayan his analizi çalışmaları yapılmıştır. Tüm öğrenme problemlerinde olduğu gibi makine öğrenmesiyle his analizi yapabilmek için çalışılan her sınıftan dengeli ve çok miktarda örneğe ihtiyaç bulunmaktadır. Ancak insanlar doğası gereği her hissi aynı oranda dile getirmemektedir. Sosyal medya gönderilerinde genellikle şikâyetler dile getirilirken memnuniyet içeren paylaşımlar daha az yapılmaktadır. Aynı şekilde insanlar paylaşımlarında ifade edilmesi daha kolay olan öfke, üzüntü ve sevinç hislerine yer verirken şaşırma ve tikslenme gibi ifade edilmesi daha güç olan hislerden pek bahsetmemektedir. Bu durum ihtiyaç duyulan öğrenme örneklerinin dengesiz dağılmasına ve belirsizliğin artmasına sebep olmaktadır. Bu belirsizlik öğrenme yöntemlerinin bazı his sınıflarını doğru şekilde öğrenmesini zorlaştırmaktadır. Bu tezde çözülmesi amaçlanan ilk problem belirsizliğin yüksek olduğu sınıflarda daha etkili ve yüksek doğrulukta Türkçe his analizi yapmaktır. Bu problemin çözümü için derin öğrenme ve optimizasyon yöntemleri bir arada kullanılarak bir his analizi şeması önerilmektedir.

Önerilen optimizasyon destekli his şeması sayesinde görece daha az örneğe sahip his sınıflarında belirsizlik azaltılarak daha başarılı Türkçe his sınıflama yapılması amaçlanmaktadır. Geleneksel derin öğrenme yöntemleri öğrendikleri model yardımıyla, test aşamasında verilen her girdi için sınıfsal üyelik dereceleri içeren bir çıktı vektörü üretmektedir. Girdinin hangi sınıfa ait olduğuna karar verilirken bu üyelik derecelerinden en yükseği seçilerek sınıf ataması yapılmaktadır. Ancak his verileri doğası gereği birçok belirsizliğe sahiptir ve en yüksek dereceyi seçmek gibi doğrusal yöntemler bu belirsizliği gidermek için yetersiz kalmaktadır. Bu tezde önerilen şema, optimizasyon yöntemleriyle üretilen optimum his vektörleri yardımıyla bu belirsizliği azaltarak daha başarılı Türkçe his sınıflama yapmayı amaçlamaktadır. Ayrıca mevcut derin öğrenme yöntemlerinin karar mekanizmasında yapılan bu değişiklik dile bağlı öğeler içermediğinden, önerilen yaklaşım dilden bağımsızdır. Bu sebeple Türkçe için önerilen yaklaşım karşılaştırma yapmak amacıyla İngilizce veri kümelerinde de denenmiştir. Öte yandan yöntemlerin

başarısı ölçülürken kullanılan genel doğruluk metriği, görece çok örneğe sahip olan büyük sınıfların başarısıyla doğru orantılıdır. Bu tez kapsamında dengesiz dağılım gösteren veri kümelerinde yer alan tüm sınıfların başarısını göstermek için sınıf içi performanslar kullanılmaktadır.

Önerilen şema farklı sınıfsal dağılımlar gösteren değişik veri kümeleri üzerinde denenmiştir. Önerilen yöntemle elde edilen sonuçlar geleneksel derin öğrenme yöntemlerinden alınan sonuçlarla karşılaştırılmış ve tablolar halinde sergilenmiştir. Karşılaştırma sonucunda, veri kümelerinin özellikle belirsizliğin yüksek olduğu görece küçük sınıflarında sınıf içi performansın bariz şekilde arttırıldığı belirlenmiştir. Ayrıca bu artış büyük sınıfların sınıf içi performanslarını olumsuz etkilememektedir. Bu sayede, optimizasyon destekli his sınıflama yöntemiyle belirsizliğin yüksek olduğu örnek uzayında daha etkili ve yüksek doğrulukta Türkçe his analizi başarılı şekilde gerçekleştirilmiştir.

Öte yandan geleneksel makine öğrenmesi yöntemleri girdi olarak ham metin yerine öznitelikler kullanılmaktadır. Bu öznitelikler metinlerden elde edilen kelime çantası (Bag of Words - BOW) [1], kelime n-gramları [2], noktalama işaretleri, emojiler ve his sözlüklerinden elde edilen öznitelikler[3] olmaktadır. Ayrıca girdinin kalitesini arttırmak ve boyutunu azaltmak amacıyla bu öznitelikler üzerinde seçme, ağırlıklandırma ve çıkarma gibi öznitelik mühendisliği işlemleri yapılmaktadır. Gerçekleştirilen bu öznitelik mühendisliği işlemleri metin içerisinde yer alan bazı önemli bilgilerin yok olmasına ve sınıflama başarısının düşmesine sebep olmaktadır. Ayrıca girdilerin temsil edildiği BOW gibi yöntemler cümle içerisinde yer alan kelimelerin sırasının ve bütünlüğünün kaybolmasına sebep olmaktadır.

Geleneksel makine öğrenmesi yöntemlerinde karşılaşılan bu olumsuzlukları ortadan kaldırmak amacıyla derin öğrenme yöntemleri kullanılmaktadır. Derin öğrenme yöntemleri metni ham haliyle girdi olarak almakta ve bir arada geçen kelimelerin bağlamını korumaktadır. Bu sayede geleneksel makine öğrenmesi yöntemlerine göre daha başarılı his sınıflama yapılmasına olanak sağlanmaktadır. Öte yandan derin öğrenme

yöntemleri birçok gizli katmanı bulunan katmanlar arasında ağırlıkların her iki yönde güncellendiği bir tür yapay sinir ağlarıdır. Bu katmanlar arası ağırlıkların optimize edilmesiyle bağlam belirlenirken kıymetli öznitelikler ön plana çıkarılarak his sınıflama modeli oluşturulmaktadır. Ancak derin öğrenme yöntemlerinde kaliteli bir model ortaya çıkarabilmek için büyük veriye ihtiyaç duyulmaktadır. Oysa Türkçe his analizi konusunda etiketlenmiş veri miktarı oldukça azdır. Bu sebeple derin öğrenme yöntemleriyle oluşturulan his sınıflama modellerinin başarısı kısıtlanmaktadır.

Derin öğrenme yöntemlerinde ihtiyaç duyulan büyük veri problemini çözmek amacıyla ortaya atılan kelime özyerleşikleri ve ön eğitilmiş dil modeli yöntemleri, doğal dil işleme (DDİ) alanlarındaki araştırmaların yönünü değiştirmiştir. Bu yöntemlerin derin ağlarda kullanımları sayesinde eğitim sıfırdan başlamamakta bu sayede modelin genelleşmesi ve problemi kavraması hızlanmaktadır. Bu kapsamda kelime seviyesinde ön bilgi sağlayan kelime özyerleşikleri yöntemlerinin [4-6] yanı sıra cümle seviyesinde çalışan yöntemler [7,8] de geliştirilmiştir. Kelime anlamlarından daha fazla ön bilgi transfer etmeyi amaçlayan araştırmalar sonucunda genel amaçlı ön eğitilmiş dil modeli yaklaşımları [9,10] geliştirilmiştir. Ön eğitilmiş dil modelleri metin içerisinde yer alan kelime anlamlarının ötesinde semantik, sentaktik ve dil bilgisi kurallarının yanı sıra kullanım alışkanlıklarını öğrenmeyi hedeflemektedir. Ön-eğitilmiş dil modelleri ile derin öğrenme yöntemlerinde ihtiyaç duyulan büyük miktardaki his verisi ihtiyacı ön bilgi transferi sayesinde azalmaktadır.

Ön eğitilmiş modelleri sayesinde az örnek bulunan görevlerde bile yüksek doğrulukta his analizi yapmanın önü açılmıştır. Bu yaklaşım İngilizce his analizi problemi çözümü için kullanılmış [11] ve Türkçe haricinde farklı bazı diller için de kullanılmaya çalışılmaktadır. Bu tez kapsamında çözüm önerilen ikinci problem ise az miktarda veri üzerinde ön eğitilmiş dil modeli kullanarak daha başarılı Türkçe his analizi gerçekleştirmektir. Bu çalışmada ilk kez ön eğitilmiş dil modelleri kullanılarak Türkçe metinler üzerinde his sınıflandırması gerçekleştirilmiştir. Bidirectional Encoder Representations from Transformers (BERT) [9] ve Türkçe için geliştirilen BERTurk[12] ön eğitilmiş dil modelleri Türkçe his sınıflandırma problemine adapte edilmiştir. Bu iki model ile birlikte, geleneksel makine öğrenmesi ve derin öğrenme yöntemleri

karşılaştırmalı olarak Türkçe veri kümeleri üzerinde uygulanmış ve elde edilen sonuçlar karşılaştırmalı olarak incelenmiştir. Bu tez kapsamında ön eğitilmiş dil modelleri kullanılarak geliştirilen Türkçe his sınıflama yönteminin hem alan yazında raporlanan sonuçlardan hem de diğer yöntemlerle elde edilen sonuçlardan daha yüksek performans sağladığı görülmüştür. Bu sayede, ön eğitilmiş dil modelleri kullanılarak geliştirilen yöntemle az veri bulunan Türkçe için yüksek doğrulukta his analizi başarılı şekilde gerçekleştirilmiştir. Devamında ise önerilen iki yöntem birlikte çalıştırılarak daha başarılı his analizi gerçekleştirilmiştir.

Bu tez kapsamında Türkçe yüksek performanslı his analizi yapılması amaçlanmaktadır. Bu amaçla hem belirsizliğin yüksek olduğu dengesiz dağılıma sahip olan veriler için hem de az öğrenme verisi bulunan durumlar için çözümler önerilmektedir.

1.2. Amaç ve Motivasyon

Bu tez kapsamında Türkçe yüksek performanslı his analizi gerçekleştirmek amaçlanmaktadır. Bu amaca ulaşmak için öğrenme kalitesini olumsuz etkileyen veriye bağlı problemler üzerine odaklanılmıştır. Ana amaca erişmek üzere birbirini tamamlayan iki alt amaç tanımlanmıştır. Bu sayede hem dengesiz veri dağılımından hem de az veri bulunmasından kaynaklı belirsizlikler azaltılarak daha başarılı bir his analizi gerçekleştirmek amaçlanmaktadır.

Öncelikle, belirsizliği yüksek ve dengesiz dağılıma sahip his veri kümeleri üzerinde Türkçe yüksek başarımlı his analizi yapmak amaçlanmaktadır. Daha çok örneğe sahip olan his sınıflarının öğrenme üzerindeki olumsuz etkisi azaltılarak daha etkin his analizi yapılması hedeflenmiştir. Önerilen yöntemin his analizinin yanı sıra diğer doğal dil işleme yöntemlerine uyarlanabilir olması da amaçlanmıştır.

Öte yandan ön eğitilmiş dil modelleri kullanılarak az veri bulunan Türkçe için diğer öğrenme yöntemlerinden daha başarılı his analizi yapmak amaçlanmaktadır. Önerilen iki yöntem birleştirilerek yüksek performanslı his analizi yapmak amaçlanmaktadır. Bu

sayede insan-bilgisayar etkileşimini daha etkin hale getirmek hedeflenmektedir. Türkçe için geliştirilen alt yapının his analizi haricindeki doğal dil işleme problemlerine bir örnek teşkil etmesi de amaçlar arasındadır.

1.3. Özgün Değer

Bu tez, geliştirilen optimum his vektörleri yöntemi ve oluşturulan Türkçe ön eğitilmiş his modeli sayesinde aşağıda sıralanan özgün değerlere sahiptir.

- Belirsizliği yüksek ve dengesiz dağılıma sahip his veri kümeleri üzerinde dilden bağımsız olarak daha başarılı his analizi yapılması.
- Derin öğrenme yöntemleriyle birlikte optimizasyon algoritmalarının his analizi özelinde ilk kez kullanılıyor olması.
- Dengesiz dağılıma sahip veri kümesi sınıflarının başarısı ölçülürken genel doğruluk metriği yerine sınıf içi performans metriğinin kullanılması.
- Türkçe ön eğitilmiş dil modelleri kullanılarak diğer yöntemlerden daha başarılı his analizi yapılması.
- Türkçe his analizi problemi özelinde ön eğitilmiş dil modeli kullanılarak yapılan ilk çalışma olması.

1.4. Tez Organizasyonu

Tezin ikinci bölümünde alanyazın özeti kapsamında his analizine kullanılan temel his modelleri incelenmiştir. Devamında his analizinde kullanılan geleneksel makine öğrenmesi yöntemleri, derin öğrenme algoritmaları, ön eğitilmiş kelime özyerleşikleri, ön eğitilmiş dil modelleri ve optimizasyon algoritmalarından bahsedilmiştir. Günlük hayatta his analizi kullanımları kısaca özetlenmiştir. Devamında ise alan yazın özeti tablo halinde verilerek his analizinin tarihsel gelişimi anlatılmıştır.

Üçüncü bölümde yapılan çalışmalardan ilki olan his analizinde derin öğrenme yaklaşımlarının sürü zekası algoritmalarıyla optimizasyonu çalışması anlatılmıştır. Yöntem içerisinde gerçekleştirilen optimum his vektörü oluşturma aşamaları ayrıntılı şekilde ifade edilmiştir. Devamında deneylerde kullanılan veri kümelerinin karakteristik

özellikleri istatistiksel bilgilerle beraber verilmiştir. Deney sonuçlarını tartışırken kullanılan performans metrikleri formülleriyle birlikte tarif edilmiştir. Elde edilen deney sonuçları tablolar halinde verilmiş ve karşılaştırmalı olarak tartışılmıştır.

Dördüncü bölümde çalışmalardan ikincisi olan ön eğitilmiş dil modelleri ile Türkçe his analizi anlatılmıştır. Çalışmada kullanılan yöntem tüm ayrıntılarıyla anlatılmıştır. Çalışmada kullanılan Türkçe his veri kümeleri karakteristik ve istatistiksel özellikleriyle birlikte verilmiştir. Deney sonuçları tablolar halinde verilmiş ve karşılaştırmalı olarak tartışılmıştır.

Beşinci bölümde çalışmalardan elde edilen sonuçlar tartışılmıştır. Özgün katkılar vurgulanmış ve gelecek çalışmalardan bahsedilmiştir.

2.ALAN BİLGİSİ VE ALAN YAZIN ÖZETİ

His analizi çalışmalarında kullanılan model, yaklaşım ve teknikler gruplar halinde verilmiştir. Bu tez kapsamında kullanılan yaklaşım ve modeller daha ayrıntılı şekilde anlatılmıştır. Alandaki öncü çalışmalardan başlanmış günümüz araştırmalarına kadar tarihsel sıra gözetilerek anlatılmıştır. His analizinde kullanılan temel his modelleri, his sınıflama algoritmaları, ön eğitilmiş dil modelleri, optimizasyon algoritmaları, Türkçe his analizi çalışmaları ve his analizinin kullanım alanları alt başlıklar şeklinde sıralanmıştır.

2.1. His Modelleri

Bilgisayarlı his analizi araştırmalarında evrensel temel his sınıfları kullanılmaktadır. Hislerin sınıflandırılması ve temel hislerin belirlenmesi için psikoloji ve sosyoloji alanında birçok çalışma yapılmıştır. Tüm çalışmalar, iç içe girmiş his evreninde yer alan temel bileşenleri ve tetikleyicileri belirlemeye çalışmaktadır. Birbirinin tamamlayıcısı şeklinde tarif edilebilen çalışmalar hala güncelliğini korumakta ve güncel çalışmalar tarafından atıf almaya devam etmektedir. Bilgisayarlı his analizi çalışmalarında genellikle Ekman, Plutchik ve Russel olmak üzere üç temel his modeli kullanılmaktadır. Bu his modelleri aşağıda ayrıntılarıyla verilmektedir.

2.1.1. Ekman'ın Temel Hisler Teorisi

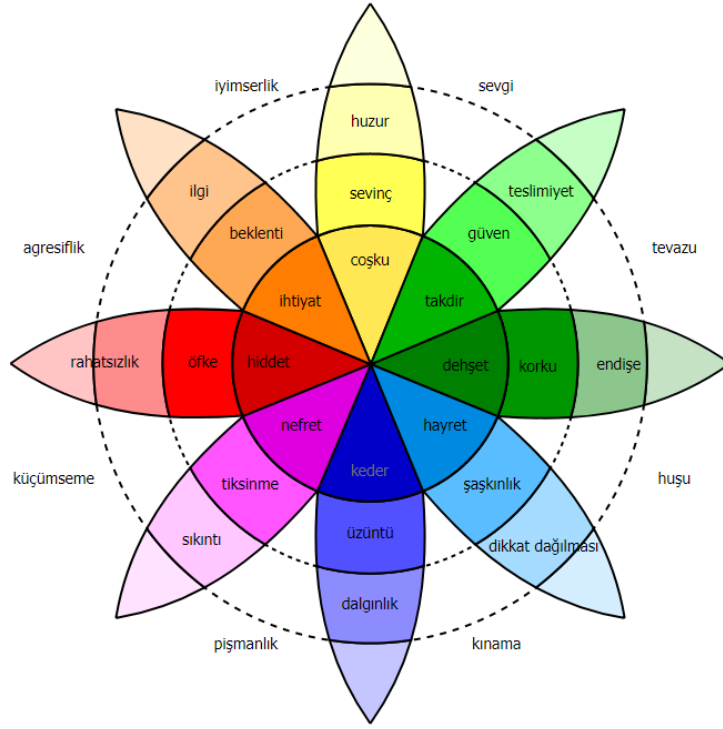
Evrensel temel his sınıflarını tespit etmeye çalışan Ekman [13,14] Darwin'in "İnsanlarda ve hayvanlarda hislerin ifadesi"[15] adlı eserinden esinlenerek dünyanın farklı bölgelerinde incelemeler yapmıştır. İncelemeler sırasında genellikle dış dünyaya kapalı kabile hayatı yaşayan insanların fotoğraflarını çekmiştir. Araştırmalar sonucunda temel hislerin yüz ifadelerinin (Şekil 2.1) öğrenilmediği, doğuştan bilindiği ve bu nedenle evrensel olduğu tespit edilmiştir. Ancak hangi durumda hangi hissin tetiklenebileceğini kültürel farkların belirlediği tespiti de eklenmiştir. Ekman[16] yaptığı devam çalışmasında temel hislerin kategorilere bölünebileceğini ve birbirine karışmayacağını öne sürmektedir. Ekman'ın "üzüntü, neşe, korku, öfke, tikslenme ve şaşırma" olmak üzere belirlediği 6 temel his kategorisi his analizi çalışmalarında sıklıkla kullanılmaktadır.



Şekil 2.1 Ekman Temel Hislerine Ait Örnek Yüz İfadeleri

2.1.2. Plutchik'in Hislerin Çarkı Teorisi

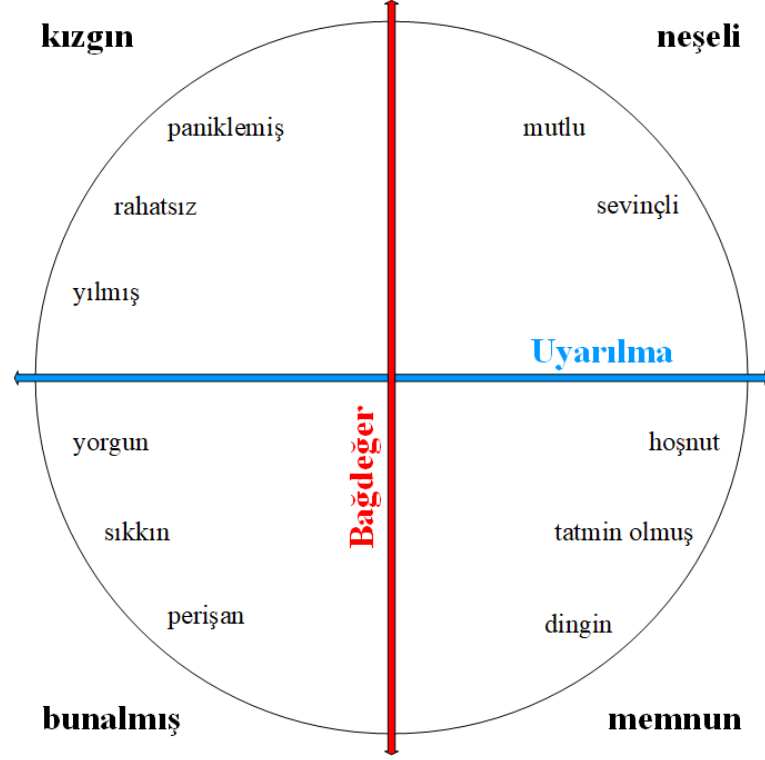
Plutchik[17] geliştirdiği his modelini bir çarkın dişlileri şeklinde tasarlamıştır. Bu çark “sevinç, güven, korku, şaşkınlık, üzüntü, tikslenme, öfke ve beklenti” temel hislerini temsil eden 8 pedaldan oluşmaktadır (Şekil 2.2). Her pedal üç katmandan oluşmaktadır. İlk katmanda sıkça bir arada bulunan hisler, ikinci katmanda bazen bir arada bulunan hisler, üçüncü katmanda ise nadiren bir arada bulunan hisler yer almaktadır. Plutchik temel hislerin yanı sıra karmaşık hisleri de sınıflandırmıştır. Plutchik his çarkı üzerinde temel hisleri temsil eden pedalların kesiştiği kısımlarda karmaşık hisler yer almaktadır. Örneğin “sevinç ve güven” hislerinin birleşiminden “sevgi” hissi elde edilmektedir. Her ne kadar karmaşık ve zıt hislerin birbirleriyle ilişkisi modelde tarif edilse de metinlerde his analizi çalışmalarında genellikle 8 temel his sınıfı kullanılmaktadır.



Şekil 2.2 Plutchik'in His Çarkı modeli

2.1.3. Russel'in Circumplex His Modeli

Temel hisler teorilerinin hisleri ifade etmekte yetersiz kaldığını düşünen Russel [18] hisleri birçok boyutta ele almak gerektiğini savunmuştur. Araştırmacı hisleri kesin sınırları belli olmayan bir koordinat düzleminde (*Circumplex*) modellemektedir. Modele göre hisler bir dairenin üzerinde birden fazla boyutta ifade edilmektedir (Şekil 2.3). Bu boyutlar “uyarılma” (*valence*) ve “değerlik” (*arousal*) olarak tanımlanmıştır. Modelde geçen uyarılma eksenini sakinlik ve heyecan aralığını, değerlik eksenini hoşnutluk ve tatsızlık aralığını temsil etmektedir. Bu boyutlara “hâkimiyet” (*dominance*) eksenini bazı çalışmalarda [19] eklenmiştir. Model hisleri kategoriler yerine koordinat düzleminde bir nokta olarak temsil ederek hiçbir kategoriye uymayan hisleri de ifade etmeyi amaçlamaktadır. Örneğin koordinat düzleminde heyecanın ve hoşnutluğun yüksek değere sahip olduğu bölgede mutluluk hissi yer almaktadır. Temel hisleri kategorik olarak ifade etmek yerine tüm hisleri çok boyutlu düzlemde ifade etmeye yarayan bu model sıklıkla his analizinde kullanılmaktadır.



Şekil 2.3 Russel'in Circumplex His Modeli

İnsan psikolojisinin derinliklerine inen ve his evrenini tanımlamaya çalışan birçok his sınıflama çalışması mevcut olsa da bahsedilen bu üç model his analizi çalışmalarında sıklıkla kullanılmaktadır.

2.2. His Sınıflama

His analizi terimi bu tez kapsamında metin verileri üzerinde DDİ araç ve yöntemleriyle yapılan his sınıflama yaklaşımlarını ifade etmek için kullanılmaktadır. His sınıflama metin içerisinde hangi hislerin ne oranda yer aldığını bulma işlemine verilen addır. His sınıflama yaklaşımları metnin bileşenlerini inceleyerek his barındıran kesimleri bulmayı hedefler. Metnin bileşenleri kelime, cümle, paragraf ve noktalama işaretleridir. Bazı durumlarda metin içerisinde yer alan his ikonları (*emoji*) da bu bileşenlerden biri sayılmaktadır.

Metin bileşenleri tek başlarına bir anlamı ve hissi ifade edebilecekleri gibi bulunduğu bağlama göre farklı anlam ve hisleri de ifade etmektedir. DDİ araçlarının anlamı kavramak için yaşadığı evrimsel süreç his sınıflama yaklaşımlarını da birinci dereceden

etkilemiştir. Bu gelişim sürecini resmetmek amacıyla his sınıflandırma çalışmaları ilkel yöntemlerden en gelişmiş yöntemlere doğru tarihsel olarak aşağıda verilmektedir.

2.2.1. Anahtar Kelime Tabanlı Sınıflama

Anahtar kelime tabanlı his sınıflama, hislerle ilgili anahtar kelimelerin metinde tespit edilmesi prensibine göre çalışmaktadır [20–24]. His sınıflarına ait anahtar kelimeler önceden bir uzman tarafından belirlenebildiği gibi bir sözlükten yararlanılabilmektedir. Çalışılan her bir his sınıfı için benzersiz kelimeler seçilmekte ve his sınıfları işaretlenmektedir. Bu yaklaşımda his sınıflama yapılacak metin parçası içerisinde yer alan anahtar kelimenin etiketi metin parçasının sınıfını belirlemek için yeteli olmaktadır. Öte yandan metin içerisinde yer alan olumsuzluk durumu, birden fazla sınıftan kelime içermeme durumu, hiçbir sınıftan kelime içermeme durumu gibi yöntemin başarısız olduğu örnekler oldukça çoktur. Öte yandan bağlama bağlı anlam kazanan kelimelerin varlığı da yöntemin büyük hatalar yapmasına neden olmaktadır. Bahsedilen olumsuzlukları sebebiyle anahtar kelime tabanlı his sınıflama yöntemi oldukça başarısız ve yetersizdir. Bu sebeple çoğu zaman kullanılmamakta veya daha gelişmiş his sınıflandırma yöntemlerine yardımcı olarak kullanılmaktadır.

2.2.2. Sözlük Tabanlı Sınıflama

İçerisinde kelimeler ve bu kelimelerin ifade ettiği hislerin yer aldığı listelere his sözlüğü adı verilmektedir [25–27]. Bu sözlükler kullanılarak yapılan his sınıflandırma yaklaşımına sözlük tabanlı his sınıflandırma adı verilmektedir.

Alanda yapılan ilk sözlük olan *General Inquirer* (GI) Stone ve ekibi [28] tarafından 1966 yılında yapılmıştır. Sözlük içerisinde “zevk, acı, duygu ve uyarılma” ve diğer kategorilerde toplam 11.788 kelime bulunmaktadır. Anahtar his kelimeleriyle anlamsal olarak ilişkili terimlerin kullanıldığı bir çalışma olan *WordNet Affect Lexicon* (WAL) sözlüğü [29] içerisinde Ekman temel hisleriyle [16] ilişkili 1.536 kelime bulunmaktadır. Mohammad ve Turney’in insan eliyle oluşturdukları NRC kelime-his ilişkileri sözlüğünde [25] Plutchik’in 8 temel his sınıfında etiketlenmiş 14.182 terim bulunmaktadır. Araştırmacılar NRC kelime-his ilişkileri sözlüğünü birçok dil için otomatik olarak çeviri yapmış ve kullanıma sunmuştur. Mohammad tarafından yapılan

NRC VAD sözlüğünde [30] ise Russel koordinat düzlemine göre etiketlenmiş 20.000 terim bulunmaktadır. Mohammad ve Kiritchenko tarafından twitter etiketlerinden faydalanarak otomatik olarak oluşturulan NRC Hashtag his sözlüğü [31] içerisinde Plutchik 8 temel his sınıflarına göre etiketlenmiş 16.862 terim bulunmaktadır. Toçoğlu ve Alpkoçak'ın Türkçe his veri kümesi[32] üzerinden oluşturdukları Türkçe His Sözlüğü [33] içerisinde Ekman temel hislerine göre etiketlenmiş yaklaşık 6.000 Türkçe kelime gövdesi bulunmaktadır.

His sınıflandırmada sözlükler bazen tek başına kullanılmakta [33,34] bazen de öznitelik çıkarma[35,36] amacıyla kullanılmaktadır. Öte yandan bağlama özel otomatik his sözlüğü oluşturma yöntemleri[26,37,38] ile probleme özel sözlükler oluşturulmuş ve geleneksel makine öğrenmesi yöntemleriyle hibrit şekilde kullanılmıştır.

His sözlüklerinde genellikle kelimeler ve bu kelimelerin yansıttıkları hisler yer almaktadır. Ancak sözlükte yer alan birden fazla anlama sahip kelimeler bağlama bağlı olarak anlam kazanmakta ve yansıttığı hisler değişmektedir. Örneğin “hastalığı kısa sürede atlattım” cümlesinde yer alan kısa kelimesi sevinci ifade ederken, “acaba boyu kısa mı kalacak?” cümlesinde ise endişeyi ifade etmektedir. Bu soruna çözüm öneren WordNet-Affect Lexicon (WAL) [29] gibi his sözlükleri kelimelerin farklı anlamları için ayrı girdilere sahiptir ve her anlam farklı hisleri yansıtabilmektedir. Ancak bu sözlükleri kullanabilmek için hedef metinde geçen her kelimenin hangi anlamda kullanıldığı bilmek gerekmektedir. Kelime anlam bilgisi büyük belirsizlikler içerdiğinden bu belirsizlikleri gidermek için ayrı çözümlere ihtiyaç duyulmaktadır. Bu durum, doğası gereği belirsizlik içeren his analizinin daha da karmaşık hale gelmesine sebep olmaktadır. Bu sebeple his sözlükleri tek başlarına bir his analizi gerçekleştirmek için yetersiz olmaktadır. Genellikle his sözlükleri sadece geleneksel makine öğrenmesi yöntemlerine öznitelik çıkarmak için kullanılmaktadır. Derin öğrenme yöntemleri ve ön eğitilmiş dil modellerinin gelişimiyle geleneksel makine öğrenmesi ve his sözlüklerine olan ihtiyaç azalmıştır.

2.2.3. Geleneksel Makine Öğrenmesi Yöntemleri

His sınıflarına ait örnekler üzerinden sınıf karakteristiklerinin matematiksel olarak modellenmesi ve bu model yardımıyla daha önce görülmemiş örneklerin ait oldukları his sınıflarının bilgisayar tarafından tahmin edilmesi işlemine makine öğrenmesi

yöntemleriyle his sınıflama adı verilmektedir [39]. Makine öğrenmesi yöntemleriyle his sınıflama yaklaşımları metin sınıflama yöntemlerinden esinlenerek uygulanmaktadır. Makine öğrenmesi tekniklerinde öncelikle metin sayısallaştırılarak öznitelikler tanımlanmaktadır.

Çizelge 2.1 Metin üzerinde gövdeleme ve parçalama

	Ham metin	Gövde metin
Doküman 1	Siyah kedi eti yedi	siyah, kedi, et, yemek
Doküman 2	Köpekler et sever	köpek, et, sevmek
Doküman 3	Kedi köpek gibisiniz	kedi, köpek, gibi
Doküman 4	Atlar da köpekler gibi sadıktır	at, köpek, gibi, sadık

Metin öncelikle DDİ araçları yardımıyla gövde veya köklere dönüştürülmekte hemen ardından anlamlı parçalara ayrılmaktadır (Çizelge 2.1). Metin sınıflama problemlerinde kelimeleri sayısallaştırmak için tanımlanan kelime çantası (BOW) [40,41] yöntemi his sınıflama problemlerine de uygulanmıştır. Kelime çantası yönteminde kelimeler sayısal kimliklere dönüştürülmekte ve sayısal olarak temsil edilmektedir (Çizelge 2.2). Buna ek olarak kelime n-gramları ve noktalama işaretleri özniteliklere dönüştürülmektedir. Kelime n-gramları, bir metin içerisinde bir arada bulunma durumunu ifade etmektedir. İfadede geçen “n” ise istenen kayar pencere boyutunu tanımlamaktadır. Kelime n-gramları sayesinde kelimenin geçtiği bağlam bilgisine ve anlam bilgisine erişmek mümkündür. Ancak doküman sayısı arttıkça hesaba katılması gereken eleman sayısı fazlaca artmaktadır. Çizelge 2.2 incelendiğinde sadece 4 doküman ve 9 kelimedenden oluşan bir uzayda bile 2-gramların sayısı 9 olmaktadır.

Çizelge 2.2 Kelime çantası ve kelime n-gramları

	1-gramlar									2-gramlar								
	siyah	kedi	et	yemek	köpek	sevmek	gibi	at	sadık	siyah, kedi	kedi, et	et, yemek	köpek, et	et, sevmek	kedi, köpek	köpek gibi	at, köpek	gibi, sadık
Doküman 1	1	1	1	1	0	0	0	0	0	1	1	1	0	0	0	0	0	0
Doküman 2	0	0	1	0	1	1	0	0	0	0	0	0	1	1	0	0	0	0
Doküman 3	0	1	0	0	1	0	1	0	0	0	0	0	0	0	1	1	0	0
Doküman 4	0	0	0	0	1	0	1	1	1	0	0	0	0	0	0	1	1	1

Çoğu zaman DDİ araçları yardımıyla elde edilen kelimelere ait kök, gövde, görev gibi dile özgü bilgiler de öznitelik olarak tanımlanmaktadır. His sınıflama bağlamından ise his ikonları ve his sözlüklerinden [25,31,42–46] elde edilen ön bilgi de özniteliklere eklenerek makine öğrenmesi algoritmalarına girdi olarak verilmektedir.

Bahsedilen tüm kaynaklardan elde edilen öznitelikler makine öğrenmesi algoritmalarına girdi olarak verilebileceği gibi öznitelikler üzerinde bazı hesaplamalar yapılması da mümkündür. Bu noktada elde edilen özniteliklerden sınıflandırmaya olumsuz ya da az katkı yaptığı düşünülenler elenmektedir. Bu sayede makine öğrenmesi algoritmalarının bellek ve işlemciyi daha makul düzeyde kullanması sağlanabilmekte ve hesaplama süresi kısaltılabilmektedir. Bu işleme öznitelik seçme adı verilmektedir. Öznitelik seçme işlemi için Ki-kare, Information Gain [47] gibi istatistiksel yöntemler kullanılabilir. Aynı şekilde sınıflandırmaya olumlu etki sağlayan özniteliklerin ağırlıkları artırılıp, daha az etki sağlayanların ağırlıkları azaltılabilmektedir. Bu işleme öznitelik ağırlıklandırma adı verilmektedir. Öznitelik ağırlıklandırma işlemi için seçme algoritmaları kullanıldığı gibi TF/IDF [48] gibi terim-sınıf bazlı ağırlıklandırma algoritmaları kullanılabilir. Öte yandan ham metin içerisinde yer almayan bazı bilgiler, öznitelik çıkarımı adı verilen yöntemlerle, birtakım kural ve sözlükler yardımıyla hesaplanarak öznitelik haline getirilmektedir. Öznitelik seçme, ağırlıklandırma ve çıkarma işlemlerinin tümü öznitelik mühendisliği şeklinde adlandırılmaktadır.

Öznitelik mühendisliği işlemlerinden geçirilen ve sayısallaştırılan his verisini modellemek amacıyla makine öğrenmesi algoritmaları kullanılmaktadır. His sınıflama işlemi bir fonksiyon olarak tanımlanırsa, girdi “öğrenme örnekleri” ve çıktı “örneklerin ait olduğu sınıflar” olacaktır. Makine öğrenmesinin görevi ise girdinin ve çıktının bilindiği durumda fonksiyonu tahmin etmeye çalışmaktır [39]. Öznitelik mühendisliğine ihtiyaç duyan ve derin öğrenme algoritmaları dışında kalan öğrenme teknikleri geleneksel makine öğrenmesi teknikleri şeklinde anılmaktadır. Alan yazında Support Vector Machines (SVM) [49], Naive Bayes [50], Decision Tree [51] gibi geleneksel makine öğrenmesi algoritmaları kullanılmıştır.

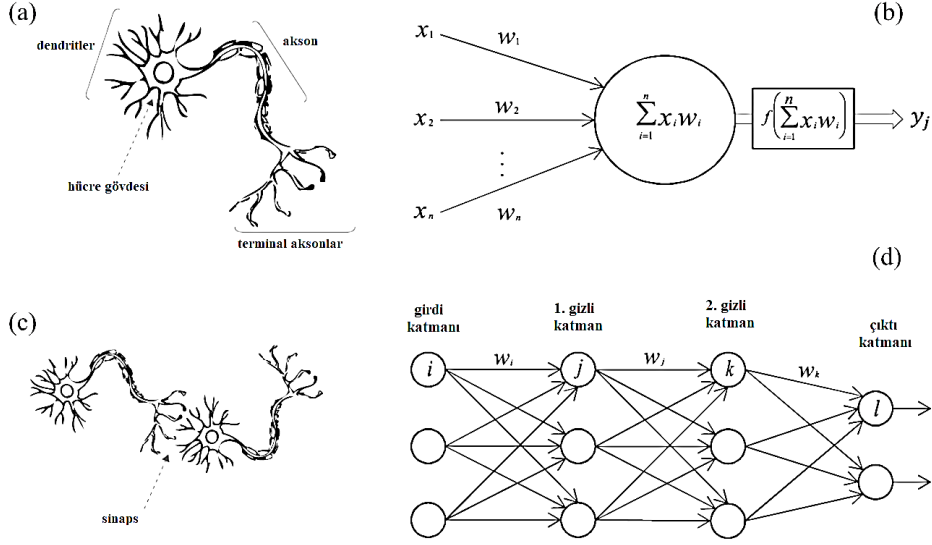
Strapparava ve Mihalcea [52] yaptıkları çalışmada 1250 haber başlığı üzerinde Ekman [16] temel his kategorilerinde Naive Bayes [50] algoritmasıyla his analizi gerçekleştirmiştir. Alm ve arkadaşları [53] yaptıkları çalışmada çocuk masallarından

alınmış 1580 cümle üzerinde SNoW [54] makine öğrenmesi yöntemiyle his analizi gerçekleştirmişlerdir. Balabantaray ve arkadaşları [55] araştırmalarında Ekman temel hislerinde etiketlenmiş 1000 tweet üzerinde SVM [49] makine öğrenmesi algoritmasıyla his analizi gerçekleştirmiştir. Roberts ve arkadaşları [56] Ekman[16] temel hislerine ek olarak sevgi (love) his sınıflarında toplam 7000 tweet üzerinde SVM [49] makine öğrenmesi algoritmasıyla his analizi gerçekleştirmiştir. Danışman ve Alpkoçak [57] ISEAR [58] veri kümesi üzerinde SVM [49], Naive Bayes [50] ve Vektör uzay Modeli ile his analizi gerçekleştirmiştir. Khan ve arkadaşları [59] yaptıkları alanyazın araştırmalarında Decision Tree [51], SVM [49], Naive Bayes [50] ve kural tabanlı algoritmaları his ve metin sınıflama özelinde karşılaştırmıştır. Sailunaz ve ekibi [60] yaptığı alanyazın özetinde geleneksel makine öğrenmesiyle gerçekleştirilmiş çoğu çalışmayı kullanılan özniteliklere, yöntemlere ve veri kümelerine göre incelemiştir.

Geleneksel makine öğrenmesi yöntemlerinin sınıflama yetenekleri girdi olarak verilen özniteliklerin kalitesiyle doğru orantılıdır. Sınıfların belirlenmesinde seçicilik özelliği yüksek öznitelikler girdiye eklendikçe sınıflama başarıları artma eğilimindedir. Ancak geleneksel makine öğrenmesi algoritmaları merkezi işlem birimi ve ana bellek üzerinde çalıştıkları için verilebilecek girdiler donanım kaynaklarıyla sınırlı kalmaktadır. Elde edilebilen tüm öznitelikler girdi olarak verildiğinde ya bellek yetersiz kalmakta ya da hesaplama işlemi insan ömründen daha uzun sürmektedir. Bu donanımsal kısıtlar geleneksel makine öğrenmesi yöntemlerinin başarısının belli bir sınırdan yukarı çıkmasına engel olmaktadır. Öte yandan bir sonraki bölümde anlatılacak olan derin öğrenme yöntemleri için bir referans noktası (*baseline*) olarak kullanılmaya devam etmektedir.

2.2.4. Derin Öğrenme Yöntemleri

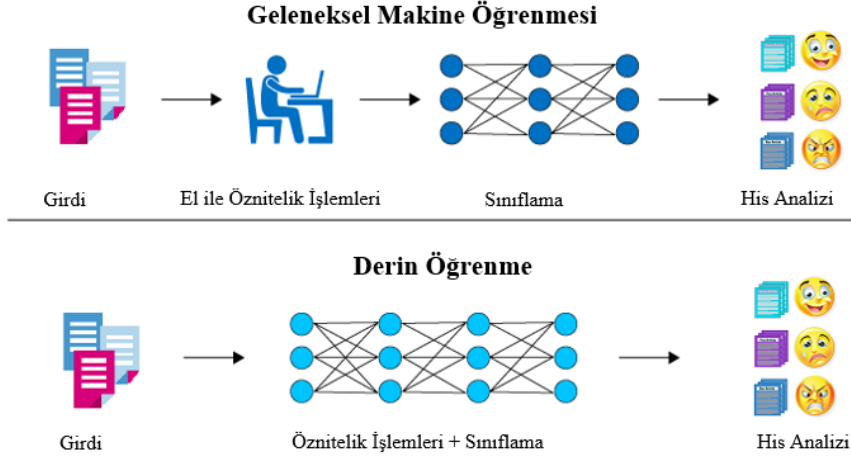
Geleneksel makine öğrenmesinin donanımsal sınıra ulaşması ve grafik kartlarında yer alan çekirdek ve bellek miktarının artarak hesaplama yapmaya uygun hale gelmesiyle son yıllarda derin öğrenme yöntemleri popüler hale gelmiştir. Aslında derin öğrenme bir tür yapay sinir ağı (YSA) algoritmasıdır.



Şekil 2.4 Doğal ve Yapay Sinir Ağları [61]

a)insan nöronu, b) yapay nöron, c) biyolojik ağ, d)YSA

Basitçe yapay sinir ağı algoritmaları; girdi katmanı, çıktı katmanı ve gizli katmanların birbirlerine belli bir fonksiyonla bağlanmasıyla çalışan algoritmalarıdır. YSA içerisinde veri bir katmandan diğerine geçerken yeni öznitelikler öğrenilir ve ağırlıklandırılır. Yapay sinir ağları insan beynindeki nöronlar arasındaki iletim ve beynin öğrenme şekline öykünerek geliştirilmiştir (Şekil 2.4). Yapay sinir ağlarının ilk ortaya atılması [62,63] 50 yıl önceye dayansa da 80'li yıllarda [64] uygulanmaya başlamış ancak bellek kapasitesi ve hesaplama gücünün yetersiz olması sebebiyle 15 yıl [65,66] öncesine kadar etkin şekilde kullanılamamıştır. Bilim insanları gizli katmanların sayısını artırarak daha fazla öznitelik çıkarmış, daha derin ağlar oluşturmuş ve derin öğrenme olarak adlandırmışlardır. Derin öğrenme uygulayan bir araştırmacı bilinen ağları kullanabileceği gibi isterse kendi ağını tasarlayabilmektedir. Geleneksel makine öğrenmesinin aksine derin öğrenme yöntemlerinde temsil (*representation*) öğrenmesi [67] yapılmaktadır.



Şekil 2.5 Makine öğrenmesi ve derin öğrenme farkları

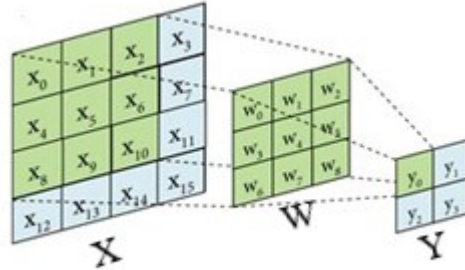
Derin öğrenmenin geleneksel makine öğrenmesinden temel farkı büyük veri üzerinde öznitelik mühendisliği yapmadan kullanılabilir olmasıdır (Şekil 2.5). Derin öğrenme kendisi hangi özneliğin önemi fazlaysa o özneliğin ağırlığını artırmaktadır. Bu sayede araştırmacının öznelikler üzerinde işlem yapmasına gerek kalmamaktadır. Ayrıca ön tanımlı kelime özyerleşikleri sayesinde geleneksel öğrenme yöntemlerinden farklı olarak metin hem anlamsal hem de söz dizimsel olarak daha başarılı ifade edilmektedir [67]. Ancak derin öğrenme yapısı oluşturulurken ağırlık yapısı, geçiş fonksiyonları, hata fonksiyonları, grup büyüklüğü (*batch size*) vb. hiper (*hyper*) parametrelerin araştırmacı tarafından optimize edilmesi gerekmektedir. Bu sebeple araştırmacı geleneksel makine öğrenmesi yöntemlerinde öznelik mühendisliği yaparken derin öğrenmede ise hiper parametre mühendisliği yapmaktadır.

Devam eden alt başlıklarda his öğrenmesinde kullanılan derin öğrenme yöntemleri tarif edilmiştir.

2.2.4.1. Evrişimli Sinir Ağları (CNN)

Yapay sinir ağlarında girdiden çıktıya doğru verinin tek yönlü iletildiği duruma ileri beslemeli ağ adı verilmektedir. Girdilerin ağırlıklarla çarpıldığı ve aktivasyon fonksiyonuna iletildiği en az bir düğümü bulunan ileri beslemeli ağlara Perceptron [41] adı verilmektedir (Şekil 4-b). Bir girdi, bir çıktı ve en az bir gizli katmanı olan ileri beslemeli ağlara ise Multi Layer Perceptron (MLP) adı verilmektedir [42]. Evrişimli Sinir

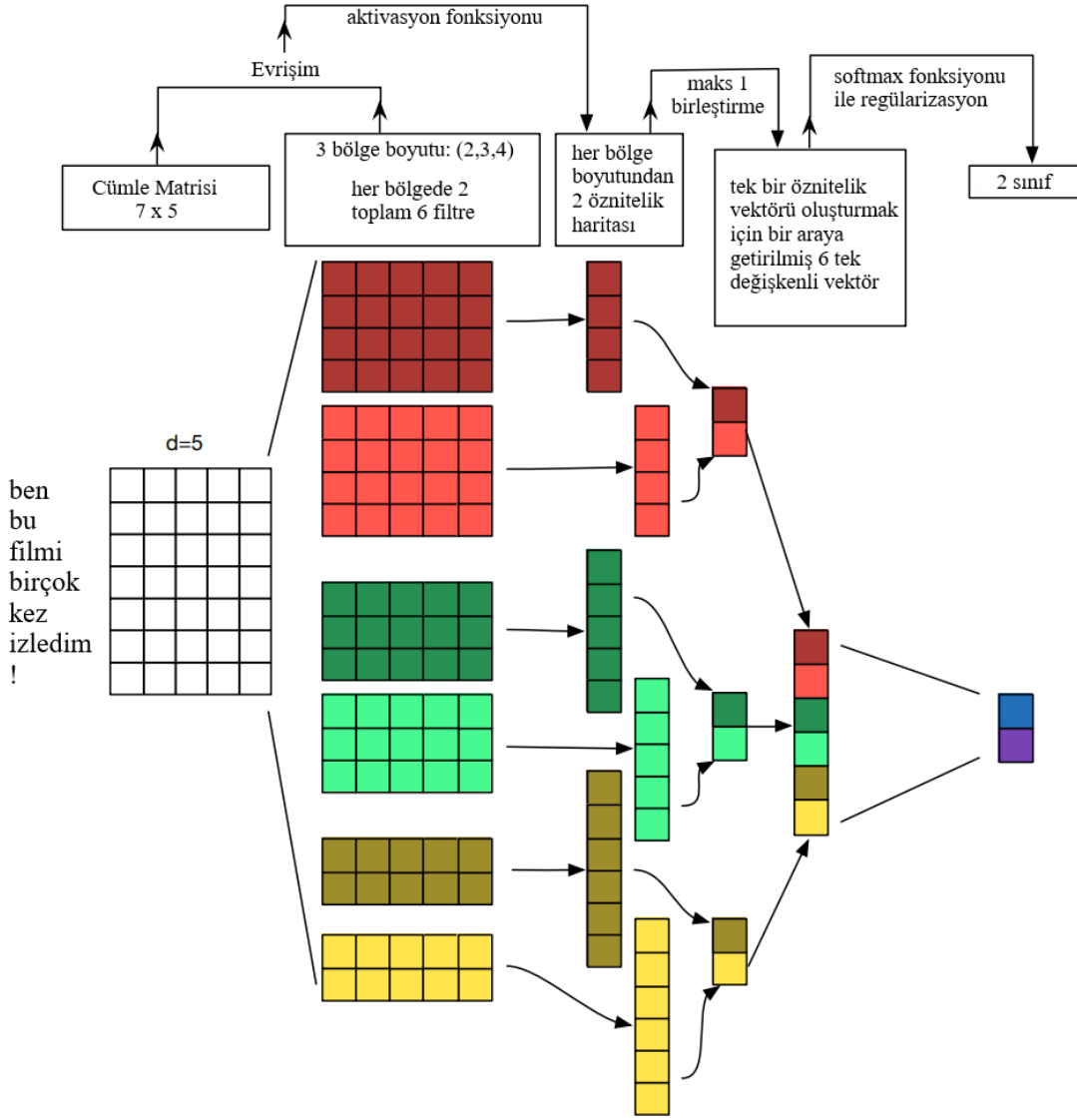
Ağları (Convolutional Neural Networks - CNN) [68] regülasyonlar sayesinde ezberleme (*overfitting*) problemine çare bulunmuş bir MLP ağıdır. En az bir katmanında normal matris çarpımı yerine evrişim (*convolution*) işlemi yapılan ağlardır.



Şekil 2.6 Evrişim (Convolution) işlemi [69]

Evrişim işleminin sembolik olarak anlatıldığı Şekil 2.6’da X (4x4 matris) “girdi matrisi”, W (3x3 matris) “filtre veya çekirdek”, Y (2x2 matris) ise “çıktı öznitelik haritasıdır”. Evrişim işleminde X üzerinde W kaydırılarak gezdirilir. Her adımda üst üste gelen elemanlar (şekilde yeşil renkte gösterilmiştir) çarpılarak toplanır ve Y üzerine yazılır.

Görüntü işleme [70] ve ses tanıma [71] alanlarında başarıyla kullanılan CNN, kelime özyerleşiklerinin geliştirilmesiyle, metin sınıflama alanında ilk kez 2014 yılında [72] kullanılmıştır. Kelime özyerleşikleri bölüm 2.2.5.1’de anlatılacağı üzere, büyük veri üzerinden gözetimsiz olarak öğrenilmiş vektörel bir kelime temsil yöntemidir. Kelime özyerleşiklerinde her kelimeye ait sabit uzunlukta bir vektör bulunmaktadır. YSA yöntemlerinde kelimeler sayısallaştırılırken kelime özyerleşikleri kullanılmaktadır. Tüm kelimeler vektörel olarak ifade edilmekte ve girdi matris şeklinde temsil edilmektedir. Geleneksel makine öğrenmesi yöntemlerinde öznitelik olarak kelime n-gramları kullanıldığında, doküman sayısı, kelime sayısı, pencere boyutu gibi değişkenlere bağlı olarak girdi matrisinin boyutu oldukça yüksek olmaktadır. CNN yönteminde ise girdinin boyutu “doküman sayısı x kelime özyerleşigi uzunluğunda” bir matris olarak sabit kalmaktadır. Öznitelik çıkarım işlemi CNN içerisinde filtreler ve evrişim işlemi ile otomatik olarak yapılmaktadır.



Şekil 2.7 CNN ile metin sınıflama yöntemi [73]

CNN ile metin sınıflama yönteminin adımlarından bahsetmek gerekirse Şekil 2.7’de görüldüğü üzere, öncelikle girdi dokümandaki kelimelerin gövdeleri bulunmaktadır. Devamında ise her bir gövde kelime özyerleşği vektörü şeklinde ifade edilmektedir. Sembolik örnekte kelime özyerleşği vektörleri 5 elemanlıdır. Girdi dokümanın kelime sayısı ise 7’dir. Girdi 7x5’lik bir matris şeklindedir. Girdi üzerinde her birinden 2’şer tane olmak üzere “2x5, 3x5 ve 4x5” boyutlarında ve başlangıçta rastgele sayılardan oluşan toplam 6 filtre ile evrişim işlemi yapılmaktadır. Bu noktada filtre sayısı ile çıkarılan öznitelik sayısı doğru orantılıdır. Ayrıca evrişim işlemi esnasında filtreler bir doğrultuda (1D) kaydırılabileceği gibi her iki yönde (2D) de kaydırılabilir. Bir sonraki katmanda ise her boyuttan iki toplam 6 tane öznitelik vektörü oluşturulmuştur. Devamındaki katmanda

ise her vektör içerisinde en büyük eleman birleştirilerek (*pooling*) 6 elemanlı bir vektör oluşmaktadır. Son katmanda ise sınıf sayısı kadar elemanı olan bir çıktı vektörü oluşmaktadır. Son katmanda yer alan vektöre bakılarak girdinin hangi sınıftan olduğuna bir aktivasyon fonksiyonu ile karar verilmektedir. Elde edilen sonuca göre katmanları birbirine bağlayan ağırlıklar güncellenir. Hata oranı en aza inene kadar bu ağırlık güncelleme işlemine devam edilir. Bu örnekte aktivasyon fonksiyonu SoftMax'tır [74].

$$S_j = \frac{e^{x_j}}{\sum_{k=1}^N e^{x_k}} \quad \forall j \in 1..N \quad (2.1)$$

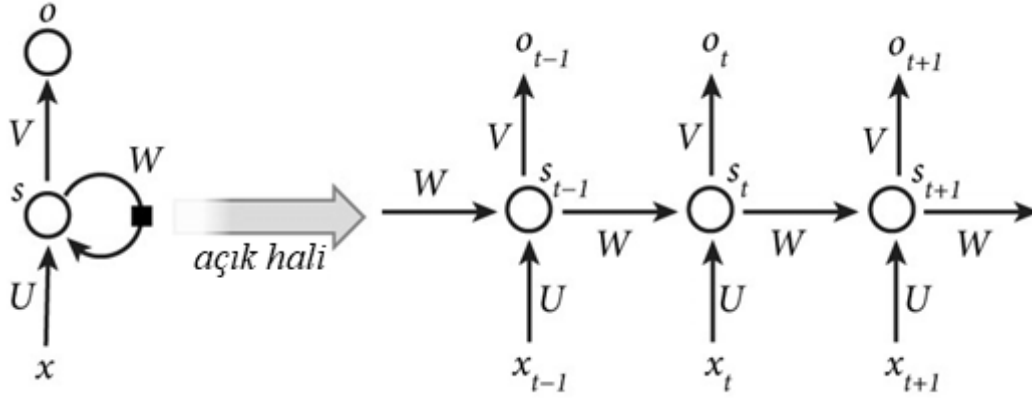
SoftMax, SoftArgMax veya normalize exponansiyel fonksiyon olarak adlandırılan aktivasyon fonksiyonu (2.1) ile girdinin hangi sınıfa ait olduğu bilgisi elde edilmektedir. Fonksiyona girdi olan vektör elemanları negatif ya da pozitif doğal sayılardan oluşmaktadır. Fonksiyon vektördeki tüm değerleri 0-1 arasına normalize ederken vektör elemanlarının toplamının 1 olmasını da garanti eder. Bu sayede girdinin sınıflara ait olma olasılıkları hesaplanabilmektedir.

Metin sınıflama için kullanılan CNN derin öğrenme yöntemi his sınıflama için de kullanılmıştır. Zahiri ve Choi [75] 7 farklı his etiketine sahip dizi replikleri üzerinde CNN derin öğrenme yöntemi ile %54 başarı sağlamıştır. Khosla [76] yaptığı çalışmada CNN yöntemi ile dizi replikleri üzerinde 4 his sınıfında %63 başarı sağlamıştır. Öte yandan CNN derin öğrenme yöntemi kelimelerin sırasını tutan bir hafızaya sahip olmadığından, devam eden bölümlerde anlatılan RNN derin öğrenme yöntemleri his sınıflama işlerinde tercih edilmektedir.

2.2.4.2. Tekrarlayan Sinir Ağları (RNN)

Tekrarlayan sinir ağları (Recurrent Neural Networks–RNN) belleğe sahip olması sayesinde ilişkili ve sıralı öğeleri öğrenebilen bir yapay sinir ağıdır. MLP ve CNN gibi İleri beslemeli sinir ağları her bir girdiyi tek başına ele alır, öncesinde veya sonrasında gelen girdi örneklerini hesaba katmaz. Oysa zaman serisi bilgileri veya cümle öğeleri gibi girdiler hep birlikte ele alınmalıdır. Çünkü cümleler kelimelerden oluşur ve kelimelerin anlamları birlikte yer aldığı diğer kelimeler sayesinde anlaşılabilir. İnsan cümlede yer alan kelimeleri teker teker okur ve cümle sonuna kadar hafızasında tutar. Cümle

sonlandığında ise içeriğin anlamını kestirmeye çalışır. Sıralı ve bağlantılı bilgileri öğrenme ihtiyacı hafızalı sinir ağlarının geliştirilmesini tetiklemiştir. Tekrarlayan sinir ağları ile konuşma tanıma [71,77], dil modelleme [78], makine çevirisi[79,80] gibi uygulamalar yapılmıştır.

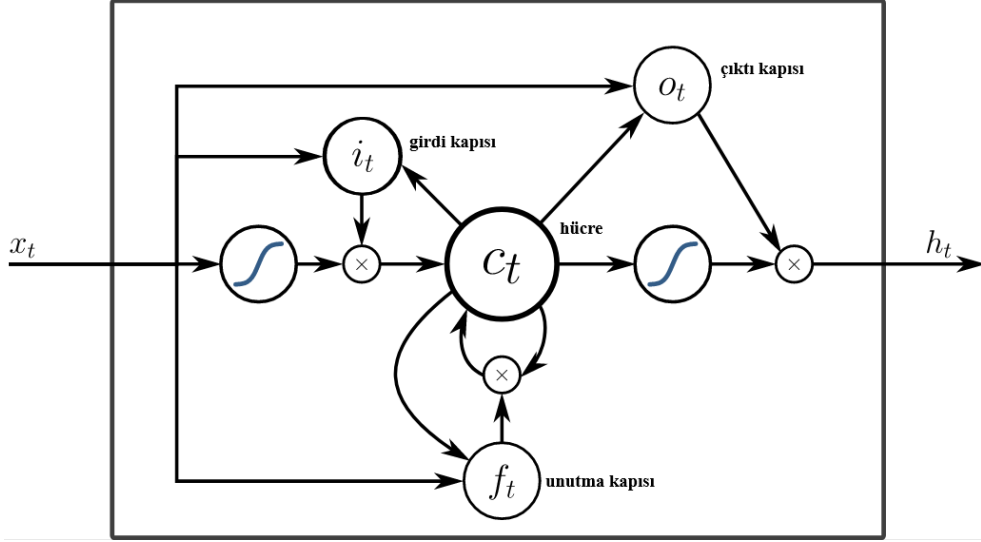


Şekil 2.8 RNN nöronunun açık hali [81]

Tekrarlayan sinir ağlarının sembolik olarak tarif edildiği Şekil 2.8’de bir RNN yapay nöronunun kapalı (soldaki) ve açık hali (sağdakiler) görülmektedir. Nöronun diğer nöronların bir önceki durumundan girdiler aldığı siyah kare ile ifade edilmektedir. Şekilde $t-1, t, t+1$ anlarındaki gizli birimler s düğümü altında gruplanmıştır. Bu yöntemle RNN, girdi sekans elemanı x_t ile çıktı sekans elemanı o_t ve tüm önceki durumlarla bağlantı kurabilmektedir. RNN ağında ağırlıklar (W) hatanın türevinin (E') çarpım işlemi ile tüm ağ üzerinde geriye doğru güncellenmektedir (*back propagation*). Bu durum RNN in ağırlıklarının çok fazla büyümesine (*explode*) veya çok küçülerek sıfırlanmasına (*vanish*) sebep olmaktadır [82]. Bu dezavantajları ortadan kaldırmak amacıyla LSTM ve GRU gibi farklı RNN türleri geliştirilmiştir. Devam eden bölümlerde bu alternatif RNN ağlarından bahsedilmektedir.

2.2.4.2.1. Uzun Kısa-Sürelili Bellek Ağları (LSTM)

RNN ağlarının dezavantajlarını ortadan kaldırmak ve daha uzun örüntüleri öğrenmek amacıyla RNN hücresi içerisinde bazı özel kapılar ve bellek hücreleri eklenerek Uzun Kısa Sürelili Bellek (Long-Short Term Memory - LSTM) ağları geliştirilmiştir [83].



Şekil 2.9 LSTM hücre yapısı [71]

LSTM hücresinin iç yapısının gösterildiği Şekil 2.9’da t anındaki hücre girdisi x_t , lojistik sigmoid fonksiyon σ , girdi kapısı i_t , çıktı kapısı o_t , unutma kapısı f_t , hücre aktivasyon vektörü c_t , hücre ile kapılar arasındaki ağırlık vektörü W , gizli vektör h ve bias vektörü b olmak üzere hücre içerisindeki hesaplamalar aşağıdaki formüller ile yapılmaktadır.

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \quad (2.2)$$

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \quad (2.3)$$

$$c_t = f_t c_{t-1} + i_t \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \quad (2.4)$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \quad (2.5)$$

$$h_t = o_t \tanh(c_t) \quad (2.6)$$

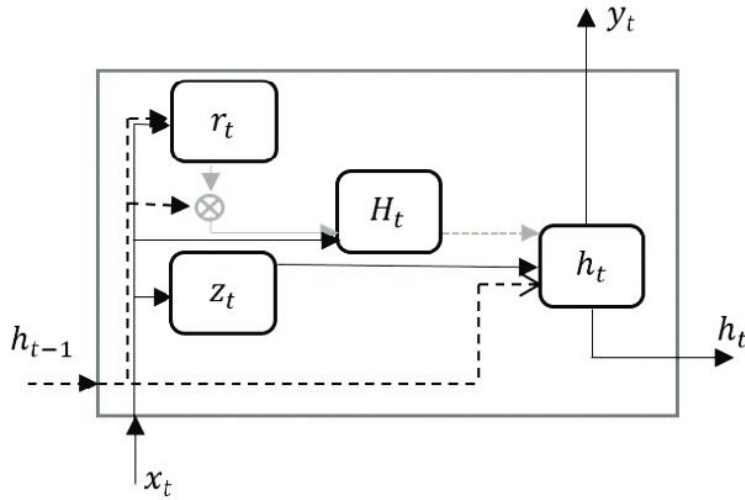
LSTM ağlarının bellekli yapısı sayesinde birbirine bağlı zincir şeklindeki bilgiler işlenebilmektedir. Bu LSTM ağlarının bağlamı yakalayabilmesine imkân tanımaktadır. Öte yandan tek yönlü LSTM ağlarının bağlam bilgisini güçlendirmek amacıyla iki yönlü (*bidirectional*) LSTM ağları (Bi-LSTM) kullanılmaktadır [84]. Bu sayede girdiler her iki yönde işlenmektedir. Metin sınıflama yöntemi olarak Bi-LSTM kullanıldığında cümlelerin kelimeleri hem baştan sonra doğru hem de sondan başa doğru ele alınmaktadır. Bu iki yönlü yapı sayesinde öğrenme kalitesi ve sınıflama başarısı artmaktadır.

Alan yazında LSTM ağları ile sıralı ve bağlantılı girdilerin öğrenilmesini gerektiren gramer öğrenme [85], el yazısı tanıma [86], konuşma tanıma [87,88] ve makine çevirisi [79] işlevleri gerçekleştirilmiştir. Bu çalışmaların yanı sıra LSTM ağlarının his analizinde

sıkça kullanıldığı gözlemlenmiştir. Köper ve ekibi [89] LSTM derin ağları ile yazarın cümleyi yazarken hissettiği his yoğunluğunu belirlemeyi hedefleyen çalışmalarında “öfke, korku, neşe ve üzüntü” his etiketlerine sahip Tweet’lerden oluşan veri kümesi üzerinde %72 başarı sağlamıştır. Naderalvojud ve ekibi his ifadelerinin gizlendiği twitter veri kümesi üzerinde [90] his özniteliklerine ek olarak Bi-LSTM ağları ile %69 başarıyla gizli hisleri belirlemiştir. Haryadi ve Kusuma [91] iç içe yerleştirdikleri LSTM ağları ile his analizi yapmışlardır. Ren ve ekibi [92] Bi-LSTM yöntemi ile çift dilli metinler üzerinde his analizi gerçekleştirmişlerdir. Gupta ve ekibi [93] geliştirdikleri semantik LSTM model ile diyaloglar üzerinde “öfke, üzüntü ve neşe” kategorilerinde his analizi gerçekleştirmiştir.

2.2.4.2.2. Geçitli Tekrarlayan Birim Ağları (GRU)

RNN ağlarının dezavantajlarını ortadan kaldırmak ve LSTM ağlarına göre daha basit bir yapı oluşturmak amacıyla 2014 yılında Cho ve ekibi [80] tarafından Geçitli Tekrarlayan Birim (Gated Recurrent Unit – GRU) yaklaşımı ortaya atılmıştır.



Şekil 2.10 Geçitli Tekrarlayan Birim yapısı [94]

GRU güncelleme kapısı z_t , ve sıfırlama kapısı r_t olmak üzere iki kapıdan oluşmaktadır. Güncelleme kapısı aktivasyon fonksiyonu (2.7) kaç birimi güncelleyeceğine, sıfırlama kapısı aktivasyon fonksiyonu (2.8) ise önceden öğrenilmiş durumları unutmasına karar vermektedir.

$$z_t = \sigma(W_z x_t + U_z h_{t-1}) \quad (2.7)$$

$$r_t = \sigma(W_r x_t + U_r h_{t-1}) \quad (2.8)$$

$$H_t = \tanh(W_H x_t + U_H (r_t h_{t-1})) \quad (2.9)$$

$$h_t = (1 - z_t)h_{t-1} + z_t H_t \quad (2.10)$$

GRU parametreleri hesaplanırken kullanılan formüller yukarıda verilmiştir. Girdi ve çıktı ağırlıkları sırasıyla W_z , W_r , W_H , U_z , U_r , U_H olarak ifade edilmektedir. Gizli katman (2.10) hesaplanırken kullanılan H_t parametresi ise (2.9) formülü ile hesaplanmaktadır.

GRU ve LSTM arasındaki temel fark ise kapı sayısı ve hücre durumlarının korunmasıdır. LSTM içerisinde “girdi, çıktı ve unutma” kapıları vardır ve dahili hafızanın durumunu korumaktadır. Bu durum LSTM ağlarını daha esnek kılsa da bellek yönetimi daha verimsizdir [95]. Bu sebeple daha az parametrenin optimize edildiği durumlarda GRU ağları tercih edilmektedir. Öte yandan GRU ağlarının bellek kapasitesi öğrenilmesi gereken uzun zincirler olduğunda yetersiz kalmakta ve LSTM ağları tercih edilmektedir [96,97].

GRU alan yazında bağlı ve sıralı öğrenme gerektiren konuşma tanıma [98], müzik sekansı öğrenme [99], motif öğrenme [100] gibi problemlere uygulamıştır. Öte yandan GRU makine çevirisi [101] ve metin sınıflama [102] problemlerinde kullanılmıştır. Abdul-Mageed ve Ungar [103] twitter verileri üzerinden etiketlerle elde ettikleri büyük his veri kümesi üzerinde GRU ağları ile Plutchik [17] temel 8 his sınıflarında %95 başarı elde etmişlerdir. Jiao ve ekibi [104] hiyerarşik GRU yöntemi ile 4 sınıflı his veri kümesi üzerinde %70 başarı elde etmişlerdir. Rathnayaka ve ekibi [105] çoklu his etiketine (*multi-label*) sahip yaklaşık 10 bin elemanlı his veri kümesi üzerinde piramit şeklinde bağlanmış GRU ağları ile his sınıflaması gerçekleştirmiştir.

LSTM, GRU ve türevleriyle gerçekleştirilen his sınıflamalarının başarıları sonraki bölümlerde anlatılan ön-eğitilmiş dil ve kelime modelleri ile bir üst seviyeye taşınmıştır.

2.2.5. Ön Eğitilmiş Modeller ile His Sınıflama

Makine öğrenmesi modelleri tarafından işlenecek metinler modelin hesaplanması için sayısal bir gösterime dönüştürülmektedir. Geleneksel makine öğrenmesi yöntemleri için

geliştirilen kelime çantası çok fazla sıfırdan oluşan seyrek (*sparse*) vektörel bir temsildir. Derin öğrenme yöntemleri bu seyrek gösterimin aksine yoğun (*dense*) vektörel bir temsile ihtiyaç duyar. Bu ihtiyacı gidermek amacıyla kelime özyerleşikleri (*word embeddings*) geliştirilmiştir. Kelime özyerleşikleri her kelime için birbirinden farklı sabit uzunlukta vektörler oluşturmak prensibine dayanmaktadır. Bu vektörler problem özelinde eğitilirken bağlama göre şekil almaktadır. Her problem için sıfırdan kelime özyerleşigi oluşturmak için çoğu zaman yeteri kadar veri bulunamamaktadır. Ayrıca her defasında kelime özyerleşigi öğrenmek için harcanan iş gücü büyük zaman almaktadır. Öte yandan bir problem özelinde öğrenilen kelime özyerleşiginin başka problemler için kullanılması oldukça güçtür. Bu dezavantajları ortadan kaldırmak amacıyla genel amaçlı ve bağlamdan bağımsız kelime özyerleşigi oluşturma fikri ortaya atılmıştır [106]. Bu kelime özyerleşikleri çok büyük veri üzerinden gözetimsiz şekilde öğrenilmiş ve çoğu probleme uyarlanabilir hale getirilmiştir.

Kelime anlamlarını yansıtan sabit uzunluktaki bu vektörel temsiller derin öğrenme algoritmasının giriş katmanında kullanılmakta ve ağıın devam eden katmanlarında problem özelinde ağırlıklar güncellenmektedir. Her problem için bu süreç tekrar edilmekte ve ağıın tüm ara katmanları sıfırdan eğitilmektedir. Kelime anlamlarını oldukça iyi yansıtan bu yaklaşım “ama, fakat, lakin” gibi anlamı tersine çeviren bağlaçlar ve olumsuzluk ekleri metin içerisinde yer aldığında yetersiz kalmaktadır. Bu ve benzeri önemli anlamsal ilişkileri modelin öğrenmesi için bu ilişkilerin yer aldığı birçok örnek üzerinden eğitim gerçekleştirilmelidir. Ancak veri kümeleri çoğu zaman yeterli sayıda örneğe sahip olmamaktadır. Doğal dil işleme alanında bu gelişmeler yaşanırken görüntü işleme yöntemleri için geliştirilen ImageNet [70] ön eğitilmiş görüntü modeli kullanılmaya başlamıştır. Bu model sayesinde eğitim sıfırdan başlanamakta ön eğitilmiş modele ait ağı yapısı ve katmanlar arası ağırlıklar bilinerek sadece sınıflama katmanı üzerinde değişiklikler yapılmaktadır. Bu önceden büyük veri üzerinden öğrenilen görüntü bilgisinin aktarımına olanak sağlamaktadır. Bu sayede derin ağı tıpkı insan gibi önceden edindiği bilgileri yeni olguları öğrenirken kullanabilmekte ve daha hızlı karar verebilmektedir. Bu ön eğitilmiş modeller kullanılarak nesne algılama[107], semantik segmentasyon [108], insan poz tahmini [109] ve video tanıma [110] gibi görüntü işleme problemleri oldukça başarılı şekilde çözülebilmektedir. Görüntü işleme alanında geliştirilen bu ön eğitilmiş modeller ve kelime özyerleşiklerin tek başına kullanılmasının

getirdiđi dezavantajlar dođal dil iřleme alanında ön eđitimi dil modeli oluřturma fikrinin oluřmasını sađlamıřtır.

Ön eđitimi dil modelleri oluřturulurken oldukça büyük veri üzerinde farklı öđrenme teknikleri ve birden fazla problemi çözmeye amacıyla oluřturulan algoritmalar kullanılmaktadır. Bu sayede genel amaçlı dil modelleri üretilmekte ve bu modellerdeki bilgi bařka alt problemlerin çözümleri için transfer edilebilmektedir.

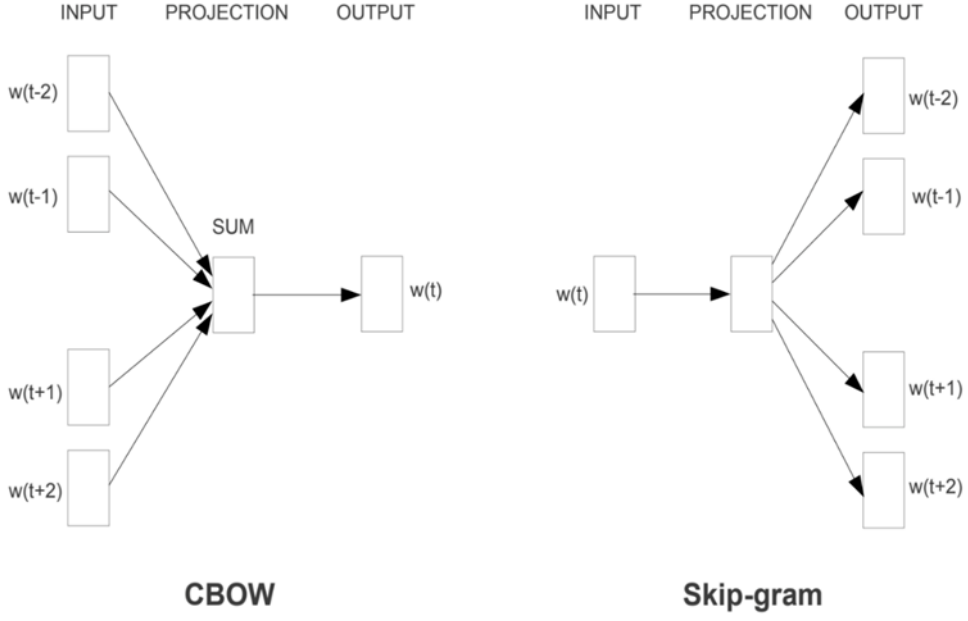
Devam eden bölümlerde kelime özyerleřiđi tekniklerinden ön eđitimi dil modellerine dođru yařanan geliřmeler özetlenmiř ve his sınıflama problemlerinde kullanımlarına deđinilmiřtir.

2.2.5.1. Ön Eđitimi Kelime Özyerleřikleri

Kelimelere ait anlamların çok boyutlu bir uzayda sabit uzunlukta yoğun vektörler şeklinde temsil edilmesine kelime özyerleřikleri adı verilmektedir. Kelime özyerleřikleri büyük veri üzerinden gözetimsiz olarak yapılan makine öđrenmesi yöntemleriyle oluřturulmaktadır. Kelime özyerleřikleri oluřturuldukları veriye, vektör uzunluklarına ve oluřturma yöntemlerine göre sınıflandırılmaktadır. Derin ađlar yardımıyla oluřturulan Word2Vec [106] ve istatistiksel yöntemlerle oluřturulan GloVe [5] en bilinen kelime özyerleřikleridir. Devam eden bölümlerde bu kelime özyerleřiklerinin yapılarından ve his sınıflama yöntemlerinde kullanımlarından bahsedilmektedir.

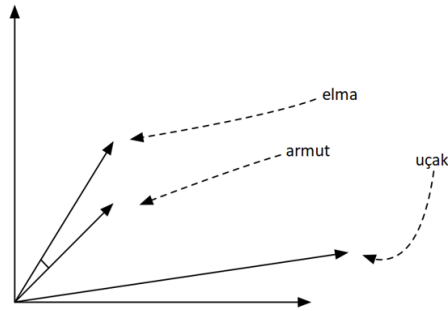
2.2.5.1.1. Word2Vec Kelime Özyerleřiđi

Word2Vec kelimelerin kelime özyerleřiđi vektörlerine dönüřtürüldüğü modeller grubudur. İki yapay sinir ađı modeli birbirinin tamamlayıcısı olarak kullanılmaktadır. Girdi olarak büyük bir derlem almakta ve her bir kelimeyi yüzlerce boyutlu anlam uzayında vektörel olarak temsil etmektedir. Anlam uzayında aynı bağlamda yakın anlamda kullanılan kelimeler birbirlerine yakın pozisyonlanmaktadır. Bu iřlem için devamlı kelime çantası (Continuous Bag Of Word – CBOW) ve devamlı Skip-gram (Continuous Skip-gram) modelleri birlikte çalıřtırılmaktadır.



Şekil 2.11 Word2Vec algoritması [106]

CBOW yöntemi örneğin kayan pencere boyutunun 5 olduğu durumda merkezdeki kelimenin w_t öncesindeki w_{t-1} , w_{t-2} ve sonrasındaki w_{t+1} , w_{t+2} kelimelerle birlikte geçme olasılığına göre tahmin edilmesidir. Skip-gram yöntemi ise kelimenin w_t bilindiği durumda kelimenin öncesindeki w_{t-1} , w_{t-2} ve sonrasındaki w_{t+1} , w_{t+2} kelimelerin tahmin edilmesidir (Şekil 2.11). Örneğin CBOW ile “kedi ___ yeri sever “cümlesinde “sıcak” kelimesini, Skip-gram ile “___ sıcak ___ ___” cümlesinde “kedi ___ yeri sever “kelimelerini tahmin etmek amaçlanmaktadır.



Şekil 2.12 Word2Vec anlam uzayında kelime vektörleri

Word2Vec ile oluşturulan anlam uzayında anlamları birbirine yakın kelimelerin vektörleri de birbirlerine yakın yerleşmektedir (Şekil 2.12). Öte yandan oluşturulan

kelime vektörleri üzerinde kelime analojisi adı verilen vektör aritmetiği işlevleri gerçekleştirilebilmektedir. Örneğin “ $V_{kral} - V_{erkek} + V_{bayan} = V_{kraliçe}$ ” “*kral*” vektöründen “*erkek*” vektörü çıkarılır ve “*bayan*” vektörü eklenirse “*kraliçe*” vektörü bulunabilmektedir. Ayrıca vektörler arası uzaklık kosinüs benzerliği (2.11) ile ölçülebilmektedir.

$$benzerlik = \cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|} \quad (2.11)$$

Word2Vec modelinin temsil yeteneği 1040 cümleden oluşan cümle tamamlama veri kümesi [111] üzerinde değerlendirilmiş ve %59 başarı sağlamıştır.

Word2Vec kelime özyerleşigi kullanılarak derin öğrenme ağlarında metin sınıflama [72], varlık tanıma [112], makine çevirisi [113], tavsiye sistemleri [114] gibi problemlere uygulanmıştır. Bunların yanı sıra farklı his sınıflama problemlerinde [115–120] kullanılmıştır.

2.2.5.1.2. GloVe Kelime Özyerleşigi

GloVe [5] tıpkı Word2Vec [106] gibi bir kelime özyerleşigidir. GloVe yöntemi ile büyük veri üzerinden kelimelerin anlamlarını yansıtan sabit uzunluklu vektörler elde edilmektedir. Bu vektörleri elde edebilmek için öncelikle tüm derlem üzerinde kelime birliktelikleri bulunmaktadır. Bu işlem yapılırken belli bir kayan pencere boyunda birlikte geçen kelimeler belirlenmektedir. Word2Vec yaklaşımında her kelime birlikteliğiyle karşılaşıldığında hesaplama yapılırken GloVe yönteminde öncelikle tüm birliktelikler sayılıp bir birliktelik matrisi oluşturulmakta ve öğrenme işlevi sonra gerçekleştirilmektedir. Oluşan büyük matrisin sütunlarında ve satırlarında kelimeler olduğunu varsayarsak, satır ve sütunların kesiştiği noktada ise o iki kelimenin bir arada bulunma sayısının bulunduğunu kabul edebiliriz. Matematiksel olarak ifade etmek için satırdaki kelime w_i ve sütundaki kelime w_j birlikte geçme sayıları ise X_{ij} olarak temsil edildiği durumda kelimelerin birlikte geçme olasılığı aşağıdaki formüllerle (2.11) - (2.12) hesaplanmaktadır.

$$X_i = \sum_k X_{ik} \quad (2.12)$$

$$P_{ij} = P(j|i) = \frac{X_{ij}}{X_i} \quad (2.13)$$

Hesaplama da kullanılan w_k kelimesi sonda (*probe*) kelimedir ve sonda kelimesinin w_i ve w_j kelimeleriyle ilişki olasılığı hesaplanmaktadır. Örneğin sonda kelimesinin “gaz” olduğu durumda “su” kelimesiyle birlikte geçme olasılığı az olurken “buhar” kelimesiyle birlikte geçme olasılığı yüksek olmaktadır. Kelime birliktelikleri hesaplandıktan sonra oluşan matris oldukça seyrek olmaktadır. Bu seyrek yapıyı yoğun hale getirmek amacıyla matris faktörizasyon işlemi gerçekleştirilmektedir. Elde edilen kelime vektörleri tıpkı Word2Vec yaklaşımındaki gibi kelimelerin anlamlarını yansıtmaktadır ve sabit uzunluktadır. Oluşturulan kelime özyerleşikleri dile, kaynak derleme ve vektör uzunluklarına göre isimlendirilmektedir.

Derin öğrenme algoritmalarının girdi katmanında kullanılan GloVe kelime özyerleşikleri yardımıyla doküman modelleme [121], varlık tanıma [122], kelime çevirisi [123] ve soru cevaplama [124] problemlerine çözüm aranmıştır. Bu çalışmalara ek olarak his analizi özelinde [125–130] birçok alt problemin çözümünde kullanılmıştır.

2.2.5.1.3. FastText Kelime Özyerleşigi

FastText [6] kelime özyerleşikleri kelimelerin yanı sıra, Word2Vec ve GloVe yaklaşımlarının aksine kelimeleri oluşturan karakter n-gramları için de birer anlamsal vektör sunmaktadır. Diğer yaklaşımlarda girdi olarak işlenen en küçük birim kelime olurken FastText yaklaşımında karakter n-gramları da hesaba katılmaktadır. Örneğin 3-4 gramların hesaplandığı modelde girdi “kavun” kelimesi olduğunda kelimenin kendisi haricinde “<ka”, ”kav”, ”kavu”, ”avu”, ”avun”, “vun”, ”vun>” ve “un>” karakter n-gramları da modele girdi olmaktadır. Yöntem olarak Word2Vec yöntemiyle aynı prensipte vektörler üretilmektedir. FastText yönteminin amacı eğitim verilerinde yer almayan ya da çok nadir kullanılan kelimeler için de bir anlamsal vektör üretmektir. İçerisinde yer almayan bir kelime sorgulandığında kelimeyi n-gramlar şeklinde böler ve model içerisinde yer alan n-gram vektörlerini birleştirerek sorgulanan kelimeyi oluşturur. Örneğin “müfris” kelimesini “müf” ve “ris” n-gramlarının bir birleşimi şeklinde ifade edebilmektedir. Bu sayede derin öğrenme yöntemine girdi olacak veri kümesi içerisinde yer alan ve özyerleşigi bulunmayan (out of vocabulary - OOV) kelime sayısını azaltmış

olur. Bu avantajlarının yanı sıra daha özyerleşik üretim esnasında daha fazla işlem zamanı ve bellek gerektirmektedir.

Derin ağlarda FastText kelime özyerleşikleri kullanılarak makine çevirisi [131], sekans etiketleme [132], hikâye üretimi [133] ve soru cevaplama [134] problemlerine çözüm üretilmiştir. Öte yandan his sınıflama problemlerinin birçoğunda [135–140] kullanılmıştır.

Kelime anlamlarını yansıtan kelime özyerleşikleri ve özyerleşikler oluşturulurken kullanılan yöntemler araştırmacıların anlamın ötesinde bazı bilgileri öğrenmenin yollarını denemesine sebep olmuştur. Önceleri probleme özel bağlamsal kelime özyerleşikleri [141] oluşturma fikri denenmiştir. Sonrasında ise anlamın ötesinde dile ait bazı diğer bilgilerin öğrenilmesini ve bu sayede dilin modellenmesini sağlamayı hedefleyen ön eğitilmiş dil modelleri devam eden bölümlerde ayrıntılı şekilde anlatılmaktadır.

2.2.5.2. Ön Eğitilmiş Dil Modelleri

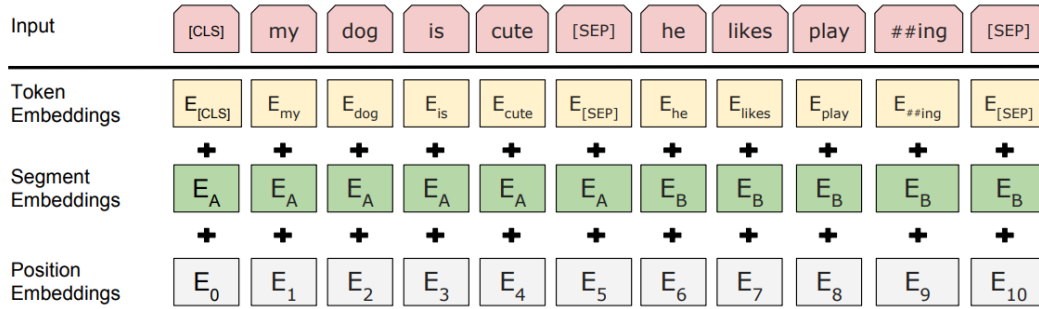
Ön eğitilmiş dil modelleri metin içerisinde yer alan kelime anlamlarının ötesinde semantik, sentaktik ve dil bilgisi kurallarının yanı sıra kullanım alışkanlıklarını öğrenmeyi hedeflemektedir. Çok büyük veri üzerinden öğrenilen evrensel dil modeli genel amaçlıdır ve birçok alt görevde kullanılabilir. Derin ağlarda ön eğitilmiş dil modeli kullanımı sayesinde eğitim sıfırdan başlamaz bu sayede modelin genelleşmesi ve problemi kavraması hızlanmaktadır. Öte yandan ön eğitim verisinin az olduğu durumlarda ezberlemeyi engelleyen bir tür düzenleme (*regularization*) aracıdır [142].

Son yıllarda yaşanan gelişmeler ön eğitilmiş dil modellerinin gelişimini tetiklemiş ve doğal dil araştırmalarının yönünü değiştirmiştir. Genel amaçlı bir dil modeli oluşturmak ve alt problemlere uyarlayarak kullanmak bir endüstri standardı haline gelmektedir. Bu ön eğitilmiş dil modellerinin öncü örneklerinden BERT [9] devam eden bölümlerde anlatılmaktadır.

2.2.5.2.1. BERT Dil Modeli

Devlin ve ekibi tarafından geliştirilen “BERT” (Bidirectional Encoder Representations from Transformers) [9] ön eğitilmiş genel amaçlı dil modelinin eğitimi için sayısal

ortamda yer alan Wikipedia gibi metin kaynaklarının yanı sıra var olan veri kümeleri dâhil edilerek 3,3 milyar kelimeyle büyük bir veri kaynağı oluşturulmuştur.



Şekil 2.13 BERT girdi gösterimi [9]

Veri kaynağından alınan metin; cümle sınırlarını, kelimeleri ve kelime konumlarını temsil eden BERT gösterimine (Şekil 2.13) dönüştürülerek algoritmaya girdi olarak verilmiştir. Girdi içerisindeki kelimelerin %15'i seçilmiş ve “Maske” anahtar kelimesiyle değiştirilerek modelin bu maske kelimeleri öğrenmesi sağlanmıştır. Modelin sadece maskeleri öğrenmek yerine tüm terimleri ve eş dizimleri öğrenmesi amacıyla, maskelenen kelimenin rastgele kelimelerle değiştirilmesi ve hiç dokunulmadan eğitime girdi olarak verilmesi gibi önlemler alınmıştır. Bu ön işlemlere tabi tutulan girdiler maskelenmiş girdiler olarak anılmaktadır. Öte yandan sonraki cümleyi tahmin etmek amacıyla ikinci cümlelerin %50'si rastgele seçilmiş cümleler ile değiştirilerek algoritmaya verilmiştir. Derin ağ 24 Transformer [143] bloğu, 1024 gizli katman ve 340 milyon parametre ile eğitilmiştir. Bu haliyle oldukça çok yer kaplayan BERT ön eğitilmiş dil modelinde yer alan bilgiler damıtılarak boyutu çok daha küçük olan DistilBERT [144] ön eğitilmiş dil modeli önerilmiştir. Bu sayede ön eğitilmiş dil modelleri daha taşınabilir ve mobil olarak kullanılabilir hale gelmiştir.

İngilizce ve diğer diller için üretilmiş ön eğitilmiş dil modelleri farklı problemlerde kullanılması amacıyla yazarlar tarafından erişime açılmıştır. Çalışma içerisinde Transformer mimarisi ile birçok alt görev üzerinde ince-ayar (*fine-tuning*) yapılan BERT ön eğitilmiş dil modeli ile başarılı sonuçlar elde edilmiştir.

BERT ön eğitimli dil modeli saldırgan ifade belirleme [145], sahte haber tespiti [146], öneri sistemleri [147], metin özetleme [148] ve birçok farklı alt probleme uyarlanarak başarılı sonuçlar elde edilmiştir.

BERT ön eğitimli dil modeli his analizi özelinde bağlamsal his yakalama [149,150], his sınıflama [151,152], diyaloglarda his sınıflama [153–155], şarkı sözlerindeki hisleri belirleme [156], twitter üzerinde his sınıflama [157], hisler ve hissi tetikleyen kelime çiftlerini bulma [158], çoklu hisleri belirleme [159] gibi birçok alt probleme uyarlanarak kullanılmıştır.

BERT [9] ön eğitimli dil modeli, Türkçe'nin de aralarında olduğu, Wikipedia içerisinde yeterli öğrenme verisi bulunan diller için çok dilli (multilingual) versiyonları bulunmaktadır. Ayrıca Türkçe için özel oluşturulan BERTurk [12] ön eğitimli dil modeli bulunmaktadır.

Bu tez kapsamında ön eğitimli dil modellerinden BERT mimarisi kullanılarak ön eğitimli bir his modeli oluşturulmaktadır (Bölüm 4). Çok dilli BERT ve BERTurk kullanılarak ince-ayar yapılan ön eğitimli his modeli sayesinde az öğrenme verisi bulunan Türkçe için daha başarılı bir his analizi yöntemi geliştirilmiştir.

2.2.5.2.2. Diğer Dil Modelleri

Ön eğitimli dil modelleri kullandığı mimariye, yapılan ön eğitim şekline, eğitim verisine, parametre sayısına ve dile göre çeşitlenmektedir. Mimariye göre LSTM [83] kullanan ELMo [160], Transformer [143] kullanan GPT [161], InfoWord [162], RoBERTa [163], XLNet [10], ELECTRA [164], UniLM [165], MASS [166], BART [167], T5 [168], ERNIE [169] sıralanabilir [142]. Öte yandan büyük bir modeli damıtarak küçülten modeller de mevcuttur. DistilBERT [144], TinyBERT [170], BERT-PKD [171], MobileBERT [172], MiniLM [173] ve DualTrain [174] BERT ön eğitimli dil modelinden damıtılarak oluşturulan modellere örnek olarak sayılabilir. Bunlara ek olarak Arapça [175], Çince [176], Fince [177], Fransızca [178], Japonca [179], Rusça [180] ve Türkçe [12] için geliştirilmiş ön eğitimli dil modelleri bulunmaktadır.

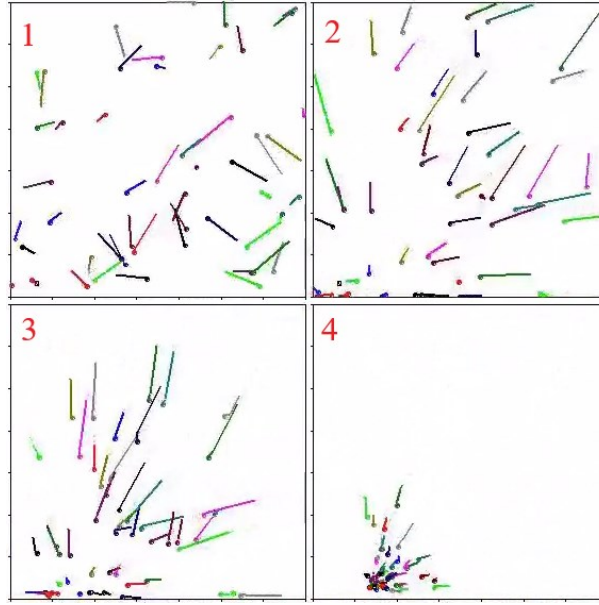
2.3. Optimizasyon Algoritmalarıyla His Sınıflama

Optimizasyon, belli koşullar altında bir probleme en uygun çözümlerin bulunmasına verilen isimdir. Matematiksel olarak ifade etmek gerekirse fonksiyonun ve çıktının bilindiği ortamda en uygun girdinin bulunmaya çalışılmasıdır. Girdi sayısı çözüm aranan uzayın boyutunu belirlemektedir. Girdi parametreleri, amaç fonksiyonu ve sınırlamalar optimizasyonun en temel bileşenleridir. Optimizasyon uygulaması, bilinen sınırlamalar altında, girdi parametreleri değiştirilerek belli bir amaç fonksiyonu çıktısının maksimize ya da minimize edilmeye çalışılmasıdır.

Optimizasyon algoritmaları deterministik ve stokastik olmak üzere ikiye gruba ayrılmaktadır. Deterministik optimizasyon algoritmaları aynı parametrelerle aynı amaç fonksiyonu her çalıştırıldığında aynı sonucu üretmektedir. Stokastik optimizasyon algoritmaları ise belli oranda rastgelelik içerdiğinden sonuç değişebilmektedir. Deterministik algoritmalar kesin en iyi sonucu, stokastik algoritmalar ise çözüme yakın en uygun çözümü ifade etmektedir [181]. Stokastik algoritmalar ise kendi içinde sezgisel ve meta sezgisel olarak ikiye ayrılmaktadır. Sezgisel optimizasyon algoritmaları belli bir problem özelinde doğruluğu kanıtlanabilir algoritmalarlardır. Meta sezgisel algoritmalar ise genellikle doğadan esinlenerek geliştirilen, sezgisel yöntemlere göre daha karmaşık ve zor problemlerin çözümünde kullanılan ve içerisinde sezgisel yöntemler barındıran optimizasyon yöntemleridir. Meta sezgisel optimizasyon algoritmalarının en sık kullanılan türü ise Parçacık Sürü Optimizasyonudur ve devam eden alt başlıklarda anlatılmaktadır.

2.3.1. Parçacık Sürü Optimizasyonu (PSO)

Parçacık sürü optimizasyonu (Particle Swarm Optimization - PSO) algoritması [182] hayvan sürülerinin avlanma ve yiyecek bulma davranışlarına öykünerek tasarlanmış algoritmalarlardır. Örneğin yiyecek arayan bir kuş sürüsünde yiyeceğin yerini bilmeyen sürü elemanları, yiyeceğe yaklaştığını öterek haber veren üyelere yakın uçma eğilimindedirler. Ava her yaklaşan üye öter ve bu döngü avın yeri tespit edilene kadar devam eder (Şekil 2.14). Sürüdeki elemanlar parçacık olarak adlandırılmaktadır.

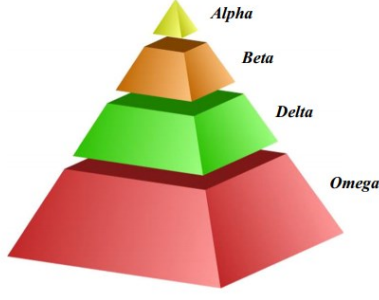


Şekil 2.14 Parçacık sürü optimizasyon örneği

Parçacıklar uzaya sınırlar dahilinde rastgele dağıtılarak optimizasyona başlanmaktadır. Devamında her bir parçacığın hedefe olan uzaklığı hesaplanmaktadır. Her parçacığın uzaydaki bir sonraki konumuna hem kendi en yakın olduğu konum hem de sürüdeki hedefe en yakın parçacığın konumu göz önünde bulundurularak karar verilir. Parçacığın en iyi değerine yerel en iyi (pBest), sürüdeki en iyi konuma ise global en iyi (gBest) adı verilmektedir. Parçacıkların hızları, konumları ve amaç fonksiyonunun maksimize edilmesi gibi hesaplamalar iterasyon sayısı kadar ve popülasyon içerisindeki her parçacığın konumu güncellenirken yapılır. Bu tez çalışması kapsamında kullanılan PSO algoritmalarından Gri Kurt optimizasyon algoritması devam eden başlıkta anlatılmaktadır.

2.3.1.1. Gri Kurt Optimizasyonu (GWO)

Gri Kurt Optimizasyon (Grey Wolf Optimization – GWO) algoritması [183] gri kurt sürülerinin hiyerarşik yapısı ve avlanma stratejilerinden esinlenerek geliştirilmiş PSO algoritmasıdır.



Şekil 2.15 Gri kurt sürüsü hiyerarşisi [183]

Gri kurt sürüsü hiyerarşisinde en baskın bireyler sırasıyla alfa (α), beta (β), delta (δ) olmak üzere 3 adettir, geriye kalan tüm bireyler omegadır (ω). Optimizasyon esnasında sürü bireylerinden en iyi konumdaki α , ikinci en iyi konumdaki β , üçüncü en iyi konumdaki δ olarak belirlenir diğer popülasyon parçacıkları ω kabul edilir. Her iterasyonda ava olan mesafe parçacığın rolünü belirler. Gri kurt optimizasyon algoritmasının matematiksel modelinde yapılan hesaplamalar aşağıdaki formüllerle verilmiştir.

$$\vec{D}_\alpha = |\vec{C}_\alpha \cdot \vec{X}_\alpha - \vec{X}_i| \quad (2.14)$$

$$\vec{D}_\beta = |\vec{C}_\beta \cdot \vec{X}_\beta - \vec{X}_i| \quad (2.15)$$

$$\vec{D}_\delta = |\vec{C}_\delta \cdot \vec{X}_\delta - \vec{X}_i| \quad (2.16)$$

$$\vec{U}_\alpha = \vec{X}_\alpha - \vec{A}_\alpha \cdot \vec{D}_\alpha \quad (2.17)$$

$$\vec{U}_\beta = \vec{X}_\beta - \vec{A}_\beta \cdot \vec{D}_\beta \quad (2.18)$$

$$\vec{U}_\delta = \vec{X}_\delta - \vec{A}_\delta \cdot \vec{D}_\delta \quad (2.19)$$

$$\vec{X}_i = (\vec{U}_\alpha + \vec{U}_\beta + \vec{U}_\delta) / 3 \quad (2.20)$$

Öncü role sahip alfa, beta ve delta parçacıklarının ava olan vektörel uzaklığı \vec{D}_α , \vec{D}_β ve \vec{D}_δ sırasıyla (2.14), (2.15) ve (2.16) formülleriyle hesaplanmaktadır. i İterasyonunda \vec{X}_i Gri Kurt'un pozisyonunu, \vec{C}_x ve \vec{A}_x katsayı vektörlerini, \vec{U}_x kurtların denedikleri pozisyonu temsil etmektedir. Gri kurt optimizasyon algoritmasının işleyişine ait sözde kod Şekil 2.16'da verilmiştir. Algoritmada parçacıklar rastgele bir konumda başlar ve sınırlar dahilinde hedefe yaklaşmak için koordineli bir şekilde hareket ederler. Maksimum iterasyon sonucunda hedefe en yakın parçacık olan alfa vektörü optimizasyon algoritmasının çıktısıdır.

```

Gri kurt popülasyonunu başlat  $X_i$  ( $i=1,2, \dots ,n$ )
 $a$ ,  $A$  ve  $C$  değerlerini başlat
Her parçacığın konumunu ve uzaklığını hesapla
 $X_a$ = en iyi konumdaki parçacık
 $X_\beta$ = ikinci en iyi konumdaki parçacık
 $X_\delta$ = üçüncü en iyi konumdaki parçacık
while ( $t <$  maksimum iterasyon sayısı)
  for  $p$  in parçacık
     $p$ 'nin konumunu güncelle (2.20)
  end for
   $a$ ,  $A$  ve  $C$  değerlerini güncelle
  Her parçacığın konumunu ve uzaklığını hesapla
   $X_a$ ,  $X_\beta$  ve  $X_\delta$  değerlerini güncelle
   $t=t+1$ 
end while
return  $X_a$ 

```

Şekil 2.16 Gri kurt algoritması sözde kodu [183]

Gri Kurt optimizasyon algoritması kullanılarak sınıflandırma da öznelik seçimi [184], görüntü segmentasyonu [185], insansız hava araçlarında yol planlama [186] gibi farklı problemlere çözüm aranmıştır. His analizi özelinde Gri Kurt Optimizasyon algoritması kullanılan bir çalışmaya rastlanmamıştır. Bu tez kapsamında ilk kez his sınıflama probleminde derin öğrenme yöntemleriyle birlikte kullanılmaktadır.

2.4. Türkçe His Sınıflama Çalışmaları

Türkçe için his analizi yapılan ilk çalışmada [187] Türkçe çocuk masallarından etiketli cümleler ve ISEAR [188] veri kümesinin çevirisi üzerinde “neşe, üzüntü, öfke ve korku” kategorilerinde 4000 örnek üzerinde his analizi yapılmıştır. Geleneksel makine öğrenimi yöntemlerinin uygulandığı çalışmanın sonucunda yaklaşık %80 oranında başarı elde edilmiştir. Türkçe diğer bir çalışmada [189] ise sosyal medya ortamlarından twitter üzerinde his analizi yapılmıştır. Kullanıcıların gönderilerinde kullandıkları etiketlere (*hashtag*) bakılarak “korku, öfke, tikslenme, neşe, üzüntü ve şaşırma” olmak üzere Ekman’ın 6 his kategorisinde toplanan her kategoride 1000 toplam 6000 tweet üzerinde geleneksel makine öğrenmesi ve sınıflama yöntemleri kullanılarak yapılan his analizi sonucunda yaklaşık %70 başarı sağlanmıştır. Başka bir çalışmada Naive Bayes kullanılarak 7 sınıfta yapılan Türkçe his analizi [190] sonucunda %86 başarı elde edilmiştir. Türkçe yapılan bir diğer çalışmada [32] araştırmacıların daha önce oluşturdukları 26000 örnek bulunan Türkçe his veri kümesi [33] üzerinden öz nitelik seçme ve ağırlıklandırma yöntemleriyle bir his sözlüğü oluşturulmuş ve bu sözlük

yardımla his analizi yapılmıştır. Ekman his kategorilerinde yapılan his analizi sonucunda yaklaşık %91 civarında başarı sağlanmıştır. Diğer bir araştırmada [191] his kategorilerine ait anahtar kelimeleri içeren Twitter gönderilerinden oluşturulan TURTED his veri kümesi üzerinde derin öğrenme algoritmalarıyla his analizi gerçekleştirmiş ve %73 başarı elde etmişlerdir. Görüleceği üzere, Türkçe his analizi alanında oldukça az çalışma yapılmış olup, çalışmaların genelde metin sınıflama çalışmalarına öykünerek yapıldığı ve sözlüklü yöntemlerin de denendiği görülmüştür.

2.5. His Analizi Uygulamaları

His analizi ürünleri ilk olarak ticari alanda kullanılmaya başlamıştır. Genellikle marka yönetimi, müşteri ilişkileri yönetimi gibi işler için kişisel bloglar, Twitter ve Facebook paylaşımları üzerinde his analizi yapılmaktadır. Öte yandan müşteri soru ve şikâyetlerini değerlendirmek için otomatik his analizi sistemleri geliştirilmiştir [192,193].

Hisler ve sanat eserleri ayrılmaz bir bütündür. Sanat eserleri sanatçının hislerini yoğun bir şekilde dışa vurduğu yapıtlardır. Metinde yer alan güçlü hislere göre müzik parçaları oluşturan bir uygulama geliştirilmiştir [194]. Öte yandan film replikleri, roman ve tiyatro eserlerinde yer alan hisleri belirleyen çalışmalar yapılmıştır [195–197].

Eğitim alanında his analizi ile otomatik özel ders ve öğrenci değerlendirme sistemleri gerçekleştirilmiştir. Sorulara verilen cevapların doğruluğunu belirlemek ve ayrıca öğrencilerin his durumu belirlemeyi amaçlayan çalışmalar mevcuttur [198,199]. Öğrencilerin sakin ve mutlu bir ortamda daha verimli öğrenebildiğini gösteren bir çalışma da yapılmıştır [200].

Sosyal medyada his akışlarını izlemek amacıyla uygulamalar yapılmıştır [201]. Deprem gibi felaketlerin insanlar üzerinde bıraktığı etkinin araştırıldığı çalışmalar mevcuttur [202]. Toplum sağlığını takip etmeyi amaçlayan bir takım uygulamalar yapılmıştır [203,204]. Siber zorbalığı tespit etmeye çalışan uygulamalar gerçekleştirilmiştir [205,206]. Dışa dönüklük veya narsisizm gibi kişilik özelliklerini kişinin ifadelerinden belirlemeyi hedefleyen psiko-sosyal çalışmalar yapılmıştır [207–209].

Toplumsal cinsiyet farklılıklarını anlamak amacıyla erkek ve kadınların kullandıkları dilin karakteristik özelliklerini inceleyen çalışmalar yapılmıştır [210]. Kadın ve erkeklerin kullandıkları dilde yansıttıkları hislerde oluşan farklar yardımıyla yazılara bakılarak yazarın cinsiyetinin tahmin edilebilir olduğu ortaya konmuştur.

Özellikle sosyal medyada paylaşılan içerikleri takip ederek toplumun politikacılara, seçimlere, ulusal değerlere olan bakış açısını belirlemeye yönelik çalışmalar yapılmıştır [211–214]. Haberlere ve sosyal medya paylaşımlarına yapılan partizan yorumları belirlemeyi hedefleyen çalışmalar yapılmıştır [215].

2.6. His Analizi Çalışmalarının Tarihsel Gelişimi

His analizi çalışmaları Çizelge 2.3 görüleceği üzere 2000’li yılların başında İngilizce için yapılmaya başlamıştır [53]. Öncelikle geleneksel makine öğrenmesi (GMÖ) yöntemleri ile his analizi çalışmaları yapılmıştır [3,53,57]. Türkçe için ilk his analizi çalışması ise 2013 yılında GMÖ yöntemleriyle yapılmıştır [187]. Türkçe için GMÖ yöntemleriyle yapılan diğer çalışma ise 2014 yılında yayınlanmıştır [190]. GMÖ yöntemlerine girdi olarak kelimelerin yanı sıra his sözlüklerinden elde edilen öznitelikler eklenerek başarı arttırılmıştır [42–45,53]. 2017 yılında derin öğrenme (DÖ) yöntemlerinin his analizinde kullanılmaya başlamasıyla sözlüklerden elde edilen ön bilgi yerini kelime özyerleşiklerine bırakmıştır [90,103,130,152]. Öte yandan Türkçe doğrulanmış ve erişilebilir bir his analizi veri kümesinin (TREMÖ) oluşturulması 2018 yılında gerçekleştirilmiştir [32]. 2019 yılında ise, TREMÖ Türkçe veri kümesi içerisinde yer alan kelime birliktelikleri istatistiklerinden yararlanarak kısıtlı bir his sözlüğü elde edilmiş ve GMÖ yöntemleriyle birlikte kullanılarak Türkçe his analizi gerçekleştirilmiştir [33]. DÖ yöntemleriyle Türkçe his analizi yaklaşımı ise ancak 2019 sonlarında yapılabilmektedir [191]. DÖ ile yapılan çalışma kapsamında Türkçe Twitter gönderilerinden oluşan TURTED his veri kümesi oluşturulmuş ve Ekman hisleri için his analizi yapılmıştır.

DÖ yöntemleriyle GMÖ yöntemlerinden ve sözlük destekli hibrit yöntemlerden daha başarılı sonuçların elde edilmesi, öğrenmeye daha fazla ön bilgi katma ihtiyacını doğurmuştur. Bu ihtiyacı gidermeyi amaçlayan ön eğitilmiş dil modeli (ÖDM)

yaklaşımlarıyla büyük veri üzerinden edinilen ön bilgiler derin öğrenme yöntemlerine aktarılmıştır.

Çizelge 2.3 Alan yazın çalışmalarının tarihsel özeti

Yıl	Referans	Dil	Yöntem	Veri Kümesi	Metrik	Sonuç%	Sözlük
2005	Alm ve ekibi [53]	İngilizce	GMÖ	Alm	F	47	WordNet
2007	Aman ve Szpakowicz [3]	İngilizce	GMÖ	Aman	Acc	74	GI + WAL
2008	Aman ve Szpakowicz[42]	İngilizce	GMÖ	Aman	F	59	WAL
2008	Danisman ve Alpkocak [57]	İngilizce	GMÖ	SE2007	F	32	
2010	Binali ve ekibi [23]	İngilizce	AKT	Blog gönderileri	Acc	96	
2010	Ghazi ve ekibi [43]	İngilizce	GMÖ	Alm + Aman	F	50	WAL
2010	Kim ve ekibi [44]	İngilizce	GMÖ	SE2007 + ISEAR + Alm	F	54	WAL + ANEW
2011	Chaffar ve Inkpen [45]	İngilizce	GMÖ	SE2007	Acc	40	WAL
2013	Perikos ve Hatzilygeroudis [24]	İngilizce	AKT	Elle oluşturulmuş	F	89	
2013	Boynukalın ve Karagöz [187]	Türkçe	GMÖ	Masallar + ISEAR çevirisi	Acc	81	
2014	Ghazi ve ekibi [46]	İngilizce	GMÖ	Aman	F	65	WAL
2014	Demirci [189]	Türkçe	GMÖ	Tweetler	Acc	70	
2014	Toçoğlu ve Alpkocak [190]	Türkçe	GMÖ	Açıcı	Acc	86	
2017	Bandhakavi ve ekibi [26]	İngilizce	GMÖ	ISEAR + SE2007+Blog	F	51	Kendi sözlüğü
2018	Toçoğlu ve Alpkocak [32]	Türkçe	GMÖ	TREMO	Acc	86	
2019	Ge ve ekibi [130]	İngilizce	DÖ	SE2019	F	75	
2019	Ma ve ekibi [152]	İngilizce	DÖ	SE2019	F	75	
2019	Basile ve ekibi [155]	İngilizce	ÖDM	SE2019	F	77	
2019	Zhong ve Miao [151]	İngilizce	ÖDM	SE2019	F	74	
2019	Xiao [150]	İngilizce	ÖDM	SE2019	F	77	
2019	Chatterjee ve ekibi [125]	İngilizce	ÖDM	SE2019	F	79	
2019	Huang ve ekibi [154]	İngilizce	ÖDM	Friends + EmotionPush	F	85	
2019	Toçoğlu ve Alpkocak [33]	Türkçe	GMÖ	TREMO	Acc	91	TEL
2019	Toçoğlu ve ekibi [191]	Türkçe	DÖ	TURTED	Acc	74	

*AKT: Anahtar kelime tabanlı, DÖ: Derin Öğrenme, GMÖ: Geleneksel makine öğrenmesi, ÖDM: Ön eğitilmiş model

Alan yazında karşılaşılan sonuçlara göre ÖDM yöntemleri kullanılarak gerçekleştirilen his analizi çalışmaları diğer tüm yöntemlerden daha başarılıdır [125,150,151,154,155]. Tez çalışması planlanırken 2017 itibarıyla, ne Türkçe his veri kümesi, ne Türkçe his sözlüğü ne de DÖ yöntemleriyle yapılmış Türkçe his analizi çalışmasına rastlanmamıştır. Bu sebeple tez kapsamında alandaki ihtiyaçları gidermek amacıyla Türkçe his sözlüğü ve DÖ yöntemleriyle his analizi gerçekleştirilmesi planlanmıştır. Ancak alandaki gelişmeler sayesinde, his analizinde sözlükler ve DÖ yöntemlerinin yerini modern ön eğitilmiş dil modelleri almıştır. Öte yandan his sözlükleri genellikle kelimeler ve o kelimelerin yansıttığı his bilgisinden oluşurken anlam bilgisi yer almamaktadır. Kelime anlamlarından bir sözlük oluşturulduğunda ise anlamı belirlemek için ayrıca kelime anlam belirsizliği giderimi yöntemlerine ihtiyaç duyulmaktadır [216]. Bu durum his verisinde doğası gereği var olan belirsizliği oldukça arttırmaktadır. Bu sebeplerle, tez çalışması kapsamında alanda daha önce denenmemiş olan optimizasyon yöntemleri ve ön eğitilmiş dil modelleriyle daha başarılı Türkçe his analizi yapılması amaçlanmaktadır.

3.OPTIMUM HİS VEKTÖRLERİ VE DERİN ÖĞRENME İLE HİS ANALİZİ

Tez kapsamında yapılan bu çalışmada, belirsizliğin yüksek olduğu dengesiz veri uzayında daha etkili ve yüksek doğrulukta Türkçe his analizi gerçekleştirilmektedir. Bu amaçla derin öğrenme yöntemleri ve optimizasyon algoritmaları bir arada kullanılarak bir his şeması önerilmektedir. Önerilen şema ile veri kümesi içerisinde yer alan sayıca baskın ya da az olan sınıfların sınıflamaya kattığı belirsizlik azaltılarak daha başarılı his analizi gerçekleştirilmektedir. Önerilen yöntem Türkçe ve İngilizce farklı his veri kümeleri üzerinde denenmiş ve sonuçları raporlanmıştır. Öte yandan referans olması amacıyla geleneksel makine öğrenmesi ve derin öğrenme yöntemleriyle deneyler tekrarlanmış ve elde edilen sonuçlar karşılaştırılmıştır.

His analizi özünde bir sınıflandırma problemi olmasına rağmen, metinlerdeki hisleri sınıflandırmak oldukça zor bir görevdir. Hisler, kelime kombinasyonlarının okuyucu üzerindeki etkisidir; dolayısıyla his analizi bu etkileri anlamayı gerektirir. Bu noktada, herhangi bir cümlenin etkisi hislerin bir karışımı olabileceği gibi yazar veya okuyucu üzerindeki gözlenen etkisi de birbirinden farklı olabilmektedir. Doğası gereği, hisler tam olarak ayırt edilemez; aralarında bazı gri bölgeler vardır. Başka bir deyişle, terimlerin farklı anlamları farklı hisleri etkileri tetikleyebilir ve his sınıfları arasındaki keskin sınırları belirlemek oldukça güçleşir. Örneğin, “korku” terimi sadece korkudan değil, aynı zamanda “üzüntü”, “tiksinme” ve şaşırtıcı derecede “sevinçten” kaynaklanabilmektedir. Buna ek olarak, cümledeki her terimden kaynaklanan belirsizliklerin toplam etkisi cümlenin his sınıfının belirlenmesini olumsuz yönde etkilemektedir. Öte yandan insanlar hislerini çoğu zaman açık bir şekilde ifade etmeyi tercih etmezler. Bu nedenle, kısa cümleler hisleri net bir şekilde ifade edemezken uzun cümleler birden fazla hissi yansıtabilmektedir. Ayrıca, eş anlamlılık, kısaltmalar, yazım hataları ve imalar belirsizliği artırmaktadır.

Makine öğrenmesi yöntemlerini kullanarak bir sınıflama problemini çözmek için; veri seti elde etme, öğrenme modeli uygulama ve genel doğruluk (accuracy) puanını bulma

aşamaları uygulanmaktadır. Özellikle son yıllarda yapılan çalışmalarda derin öğrenme yöntemleri sıklıkla kullanılmaktadır. Hem derin öğrenme hem de geleneksel makine öğrenmesi yöntemlerinde genel doğruluk, uygulanan yöntemlerin performansını ölçmek için kullanılan ana metriktir [217–222]. Bununla birlikte, his sınıflama için sadece genel doğruluk metriğine odaklanmak, belirsizlik nedeniyle yanlış anlamalara neden olmaktadır. Örneğin bir model, az elemanı bulunan bir his sınıfı için hiç doğru bir öngöründe bulunmasa bile, bu durum ilgili sınıfa ait sınırlı örneklerden dolayı genel doğruluk puanı üzerinde küçük bir etkiye sahip olmaktadır. Bu nedenle, bir his sınıflama yönteminin performansı sadece yüksek genel doğruluk üzerine odaklanmamalıdır. Aksine, düşük belirsizlik göz önünde bulundurulmalıdır. Bu noktada, sınıf içi performans dikkate alınmalıdır. Sınıf içi performans, her his sınıfı için ayrı bir tahmin başarısını ifade etmektedir. Bu tezde tartışılması amaçlanan temel nokta, sadece genel doğruluk skoruna odaklanmanın getirdiği dezavantajlardır. Özellikle, kolayca tahmin edilebilir bir his sınıfı için örnek sayısındaki artış, sınıflandırma performansını arttırmaktadır. Sonuç olarak, bir his sınıflama şemasının performansı, belirli sınıflara ait örneklerin sayısı nedeniyle tesadüfen artma veya azalma eğilimindedir. Bu durumun üstesinden gelmek için, sınıf dağılımları dengelenmeye çalışılabilir, ancak bu yaklaşım bazı hislerin dile getirilme sıklıkları nedeniyle mümkün olmayabilmektedir. Bu durumlarda az örnekli sınıflara ait sınıflama performansını doğru ölçmek amacıyla tüm his sınıfları için performansı tek tek kontrol etmek gerekmektedir.

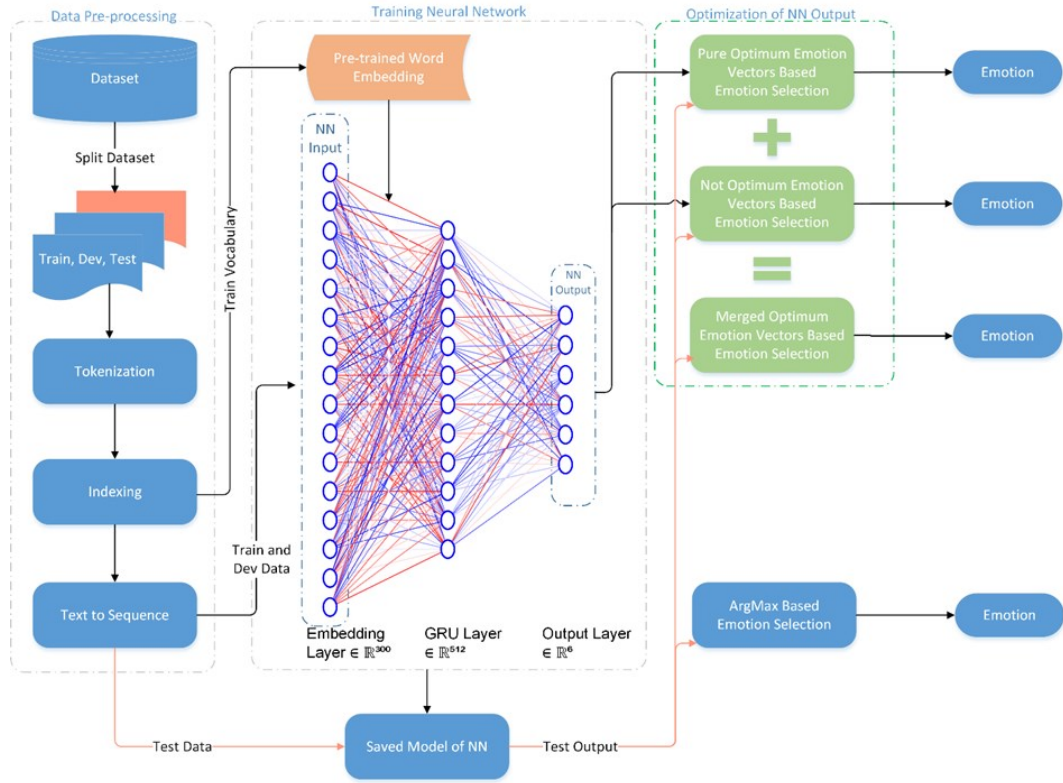
His sınıflama için derin öğrenme yöntemleri kullanıldığında, her girdi için sınıfsal olasılık oranlarından oluşan bir çıktı vektörü elde edilir. Başka bir ifadeyle, veri kümesi içerisinde toplam altı his varsa, çıktı vektörü her bir his sınıfı için hesaplanan üyelik derecelerini gösteren altı değere sahiptir. Genellikle, bu değerlerden en yüksek olanı girdinin sınıf etiketi olarak seçilmektedir. Bununla birlikte, en yüksek olasılığa sahip olanı seçmek her zaman doğru tahmini sağlamamaktadır. Bulunan olasılıklar arasında önemli bir fark olsa bile maksimum olasılığı seçmek yeterli olmamaktadır.

Maksimum üyelik derecesine göre sınıf seçmek gibi doğrusal yöntemler bağlama bağlıdır ve belirsizliği azaltmak için herhangi bir mekanizmaya sahip değildirler. Bu tez kapsamında metinlerde his analizi görevinde sınıf içi performansı arttırarak belirsizliği

azaltmak amacıyla optimizasyon temelli bir yaklaşım önerilmektedir. Bu yaklaşımda hisleri temsil eden optimum vektörler elde etmek amacıyla meta-sezgisel optimizasyon yöntemleri kullanılmıştır. Bu sayede sürü zekası optimizasyon yaklaşımlarını kullanarak her bir his sınıfı için optimum his vektörleri oluşturulmuştur. Bu kapsamda daha iyi performans elde etmek amacıyla farklı modeller denenmiş ve sonuçları karşılaştırılmıştır. Bunlara ek olarak kelimeleri temsil etmek amacıyla farklı kelime özyerleşikleri kullanılmıştır. Tüm kombinasyonlar ortalama, ortalama hataların karesi ve ağırlıklı ortalama gibi amaç fonksiyonlarla kullanılmıştır. Bu tezde, bir girdinin sınıf etiketi, o girdi için derin öğrenme modelinden elde edilen çıktı vektörü ve oluşturulan optimum his vektörleri arasındaki maksimum kosinüs benzerliğine göre belirlenmektedir. Çalışmada Gri Kurt Optimizasyon yöntemi [183] kullanılarak her his için üç farklı his vektörü üretilmiştir. Genel doğruluk değerleri yerine sınıf içi performans değerleri kullanılarak Türkçe için TREMO [32], TURTED [191] ve İngilizce için TEC [223], Tales [53], Daily Dialog [224], CrowdFlower [225] ve Blogs [3] veri kümeleri üzerinde his sınıflama gerçekleştirilmiştir. Geliştirilen yöntemle yapılan deneyler sonucunda genel doğruluk korunurken sınıf içi başarıların arttırıldığı tespit edilmiştir.

3.1.1. Yöntem

Derin öğrenme ve optimizasyon algoritmaları bir arada kullanılarak bir his sınıflama yöntemi geliştirilmiştir. Yöntemde cümlenin his sınıfını belirlemek amacıyla optimum vektörler kullanılmaktadır. Bu optimum vektörleri oluşturmak amacıyla öncelikle öğrenme kümesi üzerinde derin öğrenme modeli eğitilmektedir. Sonrasında öğrenme kümesi üzerinden elde edilen derin öğrenme modeli çıktıları optimizasyon için kullanılmaktadır. Devamında, farklı amaç fonksiyonlarıyla, derin öğrenme modelinden elde edilen çıktılar kullanılarak, her his için bir optimum vektör oluşturulmuştur. Önerilen yöntemde derin öğrenme modelinin eğitimi ve optimizasyon işlemi birbirinden bağımsız ve art arda yapılarak optimum his vektörleri elde edilmektedir. Son olarak test kümesi elemanlarına sınıf etiketi atama işlevi derin öğrenme çıktıları ve optimum his vektörleri kullanılarak gerçekleştirilmektedir.



Şekil 3.17 Önerilen optimum his vektörleri ve derin öğrenme ile his analizi yöntemi

Önerilen yöntemde Şekil 3.17’de görüldüğü üzere, öncelikle veri kümesi öğrenme, geliştirme ve test olmak üzere üçe ayrılmaktadır. Her üç parça kelimelere ayrılmakta, indekslenmekte ve sayısal olarak ifade edilmektedir. Devamında derin öğrenme yöntemi öğrenme kümesiyle eğitilmektedir. Eğitim esnasında ön-eğitilmiş kelime özyerleşikleri girdilere ağırlık olarak kullanılmaktadır. Derin öğrenme yönteminin çıktı katmanında her girdi örneği için, örneğin his sınıflarına üyelik derecelerini ifade eden vektörler oluşmaktadır. Normalde derin öğrenme yöntemi tek başına kullanıldığında girdinin sınıf etiketini çıktı vektöründeki maksimum üyelik derecesine göre (ArgMax) belirlemektedir. Öğrenme aşamasında ağ bu maksimum üyelik derecesi yaklaşımına göre eğitilmektedir. Öğrenme aşaması bittikten sonra öğrenme kümesi üzerinden elde edilen çıktı vektörleri kullanılarak üç farklı optimum his vektörü elde edilmektedir. Doğrulama aşamasında test kümesi örnekleri modele verilmekte ve modelin her bir örnek için oluşturduğu çıktı vektörleri elde edilmektedir. Test kümesi örneklerinin sınıf etiketleri belirlenirken, normalde kullanılan maksimum üyelik derecesinin kullanılması yaklaşımın aksine, oluşturulan optimum his vektörlerine olan benzerlik kullanılmaktadır. Örneğin çıktı

vektörünün en çok benzediği optimum his vektörü o örneğin sınıf etiketi olarak kabul edilmektedir. Daha açık ifade etmek gerekirse, derin öğrenme çıktı katmanı karar mekanizmasında optimizasyon temelli bir değişiklik önerilmektedir. Devam eden alt başlıkta ortaya atılan optimum his vektörlerini oluşturma aşamaları ayrıntılı olarak verilmektedir.

3.1.1.1. Optimum His Vektörleri Oluşturma

Sınıflama için kullanılan yapay sinir ağlarının çıktı katmanına Logit katmanı adı verilmektedir. Bu katman $[-sonsuz, +sonsuz]$ aralığında değerlere sahip olan logit vektörünü $X = \{x_1..x_N\}$ oluşturmaktadır. Logit vektörü SoftArgmax ya da SoftMax olarak bilinen aktivasyon fonksiyonu ile normalize edilmektedir (3.21). SoftMax fonksiyonu logit değerlerini x_i toplamları 1 olacak şekilde $[0,1]$ aralığında normalize ederek bir vektör $S = \{s_1..s_N\}$ oluşturmaktadır (3.21).

$$S_j = \frac{e^{x_j}}{\sum_{k=1}^N e^{x_k}} \quad \forall j \in 1..N \quad (3.21)$$

Burada S , çok sınıflı sınıflandırma görevlerinde tahmin edilen çıktı sınıflarına göre bir olasılık dağılımı olarak kabul edilmektedir. Bir girdinin sınıfı, o girdi için hesaplanan en yüksek üyelik derecesine sahip sınıfın etiketi ArgMax yöntemiyle tahmin edilmektedir. Bu doğrusal yaklaşımın belirsizliği giderecek bir mekanizması bulunmamaktadır. Önerilen yöntemde ise örneklerin sınıf etiketleri belirlenirken, sürü zekası optimizasyon yöntemleriyle [183] oluşturulan optimum his vektörlerine olan benzerlik kullanılmaktadır. Bu aşamada vektörler arası benzerlik kosinüs benzerlik metriğine göre hesaplanmaktadır (3.22).

$$\cos(S, C) = \frac{SC}{\|S\|\|C\|} = \frac{\sum_{i=1}^n S_i C_i}{\sqrt{\sum_{i=1}^n S_i^2} \sqrt{\sum_{i=1}^n C_i^2}} \quad (3.22)$$

Burada S (3.22) ile normalize edilmiş derin öğrenme modelinin çıktı vektörünü temsil ederken, $C = \{c_1..c_6\}$ ise sınıflardan biri için aday optimum his vektörünü temsil etmektedir. İki vektör arasındaki benzerlik $\cos(S, C)$ fonksiyonu ile hesaplanmaktadır. Girdinin sınıf etiketi, benzerliğin en yüksek olduğu his vektörüne göre atanmaktadır.

Önerilen yöntemde optimum his vektörlerini üretmek için sürü zekası yöntemlerinden biri olan Gri Kurt Optimizasyon (GWO) [183] yöntemi tercih edilmiştir. GWO gri kurtların sürü içindeki hiyerarşik yapısı ve avlanma stratejilerinden esinlenerek geliştirilen popülasyon tabanlı ve meta-sezgisel bir optimizasyon yöntemidir. Aday optimum his vektörleri (C) sürüdeki bireyleri temsil etmektedir. Arama alanındaki bireylerin konumları, her yinelemede sürüde en iyi konuma sahip olan üç bireyin konumlarına göre güncellenmektedir.

Hisleri temsil eden optimum his vektörleri, popülasyon temelli optimizasyon algoritması olan GWO ile derin öğrenme modeli öğrenme kümesi çıktıları kullanılarak oluşturulmaktadır. Bu çalışmada önerilen yaklaşımın performansını incelemek ve doğrulamak için Saf Optimum His Vektörleri (Pure Optimal Emotion Vectors – POEV), Tümlenmeyen Optimum His Vektörleri (Not Optimal Emotion Vectors – NOEV) ve Birleşim Optimum His Vektörleri (Merged Optimal Emotion Vectors – MOEV) olmak üzere üç farklı yaklaşım kullanılmaktadır.

POEV olarak adlandırılan yöntemde her his sınıfı için en ideal vektörel temsili elde etmek amaçlanmıştır. Toplam altı his sınıfı bulunduğu için her sınıf için bir POEV oluşturulmaktadır. Optimizasyon sırasında, öğrenme kümesi doğrudan kullanılmakta ve ilgili his sınıfının örnekleri POEV üretimine girdi olarak kullanılmaktadır. Her his sınıfına ait bir POEV bulmak amacıyla optimizasyon algoritması aşağıdaki denklemde ifade edilen amaç fonksiyonuna göre çalışmaktadır.

$$\text{maximize } \sum_{k=1}^K \cos(S^k, C) \quad \forall k \in 1..K \quad (3.23)$$

Denklemden (3.23) yer alan K , POEV’i hesaplanan his sınıfına ait olan örnek sayısını temsil etmektedir. S sınıf etiketi bilinen öğrenme kümesi derin öğrenme çıktısını, C ise aday POEV vektörünü ifade etmektedir.

Tamamlayıcı bir yaklaşım olarak NOEV ile diğer his sınıflarına ait olmayan örnekleri temsil eden en ideal vektör bulunmaya çalışılmaktadır. Her his sınıfına ait bir NOEV bulmak amacıyla optimizasyon algoritması aşağıdaki denklemde (3.24) ifade edilen amaç fonksiyonuna göre çalışmaktadır.

$$\text{minimize } \sum_{m=1}^M \text{Cos}(S^m, C) \quad \forall m \in 1..M \quad (3.24)$$

Denklemden (3.24) yer alan M , NOEV’i hesaplanan his sınıfına ait olmayan örnek sayısını temsil etmektedir. S ve C ise denklem (3.23) ile aynı görevdedir.

Her his sınıfı için bir POEV, bir NOEV ve POEV ile NOEV birleşimi olan bir MOEV vektörü oluşturulmaktadır. Yaklaşımında, test kümesinden gelen herhangi bir örneğin sınıf etiketi atama kuralları aşağıdaki gibi verilmiştir:

- Örneğe ait çıktı vektörü ile tüm hislere ait POEV vektörleri arasındaki benzerlik değerleri hesaplanır. Benzerlik değeri en yüksek olan POEV vektörünün sınıf etiketi örneğin sınıfı olarak kabul edilir.
- Örneğe ait çıktı vektörü ile tüm hislere ait NOEV vektörleri arasındaki benzerlik değerleri hesaplanır. Benzerlik değeri en yüksek olan NOEV vektörünün sınıf etiketi örneğin sınıfı olarak kabul edilir.
- Örneğe ait çıktı vektörü ile tüm hislere ait MOEV vektörleri arasındaki benzerlik değerleri hesaplanır. Benzerlik değeri en yüksek olan MOEV vektörünün sınıf etiketi örneğin sınıfı olarak kabul edilir.

3.1.1.2. Derin Öğrenme Modeli Uygulama Ayrıntıları

Önerilen yöntemde derin öğrenme yaklaşımı olarak Çift Yönlü Geçitli Tekrarlayan Birim Bi-GRU ağları tercih edilmiştir. GRU dâhili belleği sayesinde metin gibi sıralı verileri daha yüksek performans ile sınıflamaktadır. Çift yönlü kullanım sayesinde metinlerde yer alan bağlam her iki yönde öğrenilebilmektedir.

Deney düzeneği *TensorFlow* ve *Keras* kütüphaneleriyle Python 3.6 kullanılarak hazırlanmıştır. *Embedding*, *Bi-directional* ve *Dense* katmanları yapay sinir ağına *Keras* vasıtasıyla sırayla eklenmiştir. 300 boyutlu *FastText* ön-eğitilmiş kelime özyerleşikleri *Embedding* katmanına sabit şekilde başlangıç parametresi olarak eklenmiştir. *Embedding* katmanının ardına, 0.5 *Dropout*'a sahip 256 boyutunda bir *Bi-directional* katman eklenmiştir. *Bi-directional* katmanında GRU ve LSTM yöntemleri denenerek daha yüksek performans sergileyen GRU tercih edilmiştir. Eğitim esnasında ezberlemeyi önlemek için “*LI*” algoritması aktivite düzenleyici olarak kullanılmıştır. Son olarak, aktivasyon fonksiyonu “*Softmax*” olan *Dense* katman eklenmiştir. Modelin kayıp fonksiyonu “*Kullback – Leibler Divergence*”, optimizasyon fonksiyonu “*Nesterov Adam*” ve başarı metriği “Genel doğruluk” olarak seçilmiştir.

Denklemlerde (3.23)-(3.24) tanımlanan amaç fonksiyonları karar değişkenleri dikkate alınarak orijinal GWO üzerinde bazı düzenlemeler yapılmıştır. Popülasyon sayısı 50, yineleme sayısı 30 olarak ayarlanmıştır. Algoritma her his sınıfı ve her optimum his vektör tipi için ayrı ayrı çalıştırılmıştır.

3.1.2. His Veri Kümeleri

Alan yazında rastlanan his analizi veri kümeleri Ekman [16], Plutchik [17] ve Russel [18] temel his teorileri kategorilerine göre hazırlanmıştır. Bu veri kümeleri genellikle cümlelerden ve her cümleye atanan bir veya daha fazla his etiketinden oluşmaktadır. Bazı veri kümeleri hislerin yoğunluk oranını göstermektedir. Veri kümeleri; uzman tarafından, uzaktan öğrenme (*distant supervision*) [226] veya kitle kaynak kullanımı (*crowdsourcing*) [227] yöntemleriyle etiketlenilmektedir.

Çizelge 3.4 İngilizce his veri kümeleri istatistikleri

		Öğrenme	Geliştirme	Test	Toplam	Hisler	Kaynak	Etiket
TEC	<i>öfke</i>	1007	239	308	1554	Ekman + Pozitif şaşırma, Negatif Şaşırma	Tweetler	Uzaktan öğrenme
	<i>tiksinme</i>	495	115	151	761			
	<i>korku</i>	1801	459	550	2810			
	<i>üzüntü</i>	2462	586	775	3823			
	<i>şaşıрма</i>	2436	652	756	3844			
	<i>neşe</i>	5254	1315	1663	8232			
	Toplam	13455	3366	4203	21024			
Tales	<i>öfke</i>	457	123	152	732	Ekman	Çocuk masalları	Uzman tarafından
	<i>tiksinme</i>	253	41	84	378			
	<i>korku</i>	419	128	165	712			
	<i>üzüntü</i>	578	148	195	921			
	<i>şaşıрма</i>	515	130	161	806			
	<i>neşe</i>	1191	295	341	1827			
	Toplam	3413	865	1098	5376			
Daily Dialog	<i>öfke</i>	665	159	198	1022	Ekman	Diyaloglar	Uzman tarafından
	<i>tiksinme</i>	231	48	74	353			
	<i>korku</i>	114	24	36	174			
	<i>üzüntü</i>	724	176	250	1150			
	<i>şaşıрма</i>	1169	297	357	1823			
	<i>neşe</i>	8235	2020	2630	12885			
	Toplam	11138	2724	3545	17407			
CrowdFlower	<i>öfke</i>	927	217	277	1421	Ekman + CF hisleri	Tweetler	Kitle kaynak kullanımı
	<i>tiksinme</i>	116	21	42	179			
	<i>korku</i>	5451	1350	1629	8430			
	<i>üzüntü</i>	3304	813	1006	5123			
	<i>şaşıрма</i>	1355	367	455	2177			
	<i>neşe</i>	5958	1419	1843	9220			
	Toplam	17111	4187	5252	26550			
Blogs	<i>öfke</i>	110	26	43	179	Ekman + nötr, karmaşık hisler	Blog gönderileri	Uzman tarafından
	<i>tiksinme</i>	105	30	37	172			
	<i>korku</i>	67	19	29	115			
	<i>üzüntü</i>	105	29	39	173			
	<i>şaşıрма</i>	73	17	25	115			
	<i>neşe</i>	343	85	108	536			
	Toplam	803	206	281	1290			

Çizelge 3.5 Türkçe his veri kümeleri istatistikleri

		Öğrenme	Geliştirme	Test	Toplam	Hisler	Kaynak	Etiket
TREMÖ	<i>öfke</i>	2657	885	1181	4723	Ekman	Anket yöntemiyle	Uzman tarafından
	<i>tiksinme</i>	2036	679	905	3620			
	<i>korku</i>	2471	824	1098	4393			
	<i>üzüntü</i>	2824	942	1255	5021			
	<i>şaşıрма</i>	1689	563	751	3003			
	<i>neşe</i>	2941	980	1308	5229			
	Toplam	14618	4873	6498	25989			
TURTED	<i>öfke</i>	24834	8449	11040	44323	Ekman	Tweetler	Uzaktan öğrenme
	<i>tiksinme</i>	18001	6109	8165	32275			
	<i>korku</i>	18738	6231	8175	33144			
	<i>üzüntü</i>	19352	6292	8585	34229			
	<i>şaşıрма</i>	13359	4434	5946	23739			
	<i>neşe</i>	15653	5131	6951	27735			
	Toplam	109937	36646	48862	195445			

Önerilen yaklaşım Ekman teorisine göre etiketlenmiş veri kümeleri üzerinde denenmiştir. Önerilen yaklaşımın performans değerlendirmesi için cümlelerin sadece bir his etiketine sahip olduğu TREMO [32], TURTED [191], TEC [223], Tales [53], Daily Dialog [224], CrowdFlower [225] ve Blogs [3] veri kümeleri seçilmiştir. Çizelge 3.4’de yer alan örneklerin his kategorilerine dağılımı incelendiğinde, tiksinme ve korku olarak etiketlenen örnek sayısı oldukça düşüktür. Ancak, neşe olarak etiketlenen örneklerin sayısı diğerlerinden önemli ölçüde yüksektir. Veri kümelerindeki örnekler yazım kurallarının dikkate alınmadığı bir dille yazılmıştır. Bu olumsuz faktörler sınıflandırma performansını olumsuz etkilemektedir.

Veri kümelerindeki çok kısa ve okunaksız cümleler ile Ekman his sınıfları haricinde etiketlenmiş örnekler tespit edilerek deney dışında bırakılmıştır. Her his veri kümesi öğrenme (%64), geliştirme (%16) ve test (%20) şeklinde parçalara bölünmüştür. Bölme işlemi esnasında sınıf dağılımları korunmuştur.

3.1.3. Performans Metrikleri

Önerilen yaklaşımın performansı örneklerin dağılımındaki belirsizlik ve dengesizlik dikkate alınarak değerlendirilmiştir. Önerilen yaklaşımın performansı Kesinlik (*Precision* – P), Anma (*Recall* – R), F_1 ve Genel Doğruluk (*Accuracy*) metrikleri kullanılarak hesaplanmaktadır (3.25)-(3.28).

$$Precision(P) = \frac{TP}{TP + FP} \quad (3.25)$$

$$Recall(R) = \frac{TP}{TP + FN} \quad (3.26)$$

$$F_1 = \frac{2PR}{P + R} \quad (3.27)$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (3.28)$$

Denklemlerde yer alan TP gerçek pozitifleri, TN gerçek negatifleri, FP yanlış pozitifleri, FN yanlış negatifleri temsil etmektedir.

P , F_1 metrikleri her sınıf için ayrı ayrı hesaplanmaktadır. P ve R metrikleri, örnek sayısı sınıflar arasında eşit olarak dağıtıldığında başarı göstermektedir. F_1 dengesiz dağılımlarda daha başarılıdır çünkü F_1 , P ve R puanlarının harmonik ortalamasıdır. Bu nedenle, F_1 metriği, önerilen yaklaşımın sınıf içi performansını ölçmek için kullanılmaktadır.

Önerilen yöntemin tüm sınıflarda gösterdiği başarısını ölçmek amacıyla Makro-ortalama ve Mikro-ortalama metrikleri kullanılmaktadır. Makro-ortalama hesaplanırken sınıf büyüklükleri dikkate alınmamaktadır. Bu nedenle Makro-ortalama, küçük ve dengesiz dağıtılmış veri kümesi sınıflarındaki performansı değerlendirmek için daha uygundur. F_1 Makro-ortalama (F_{1mac}) metriğini hesaplamak için kullanılan formüller aşağıdaki gibidir (3.29)-(3.31).

$$P_{mac} = \frac{1}{N} \sum_{n=1}^N \frac{TP_n}{TP_n + FP_n} = \frac{\sum_{n=1}^N P_n}{N} \quad (3.29)$$

$$R_{mac} = \frac{1}{N} \sum_{n=1}^N \frac{TP_n}{TP_n + FN_n} = \frac{\sum_{n=1}^N R_n}{N} \quad (3.30)$$

$$F_{1mac} = 2 \frac{P_{mac} \times R_{mac}}{P_{mac} + R_{mac}} \quad (3.31)$$

Denklemlerde kullanılan N his sınıfı sayısını temsil etmektedir.

Veri kümesindeki tüm örneklerin eşit ağırlığa sahip olması tercih edildiğinde Mikro-ortalama kullanılmaktadır. Mikro-ortalama, dengesiz dağılıma sahip veri kümesi içerisindeki büyük sınıflarda gösterilen başarıyı hesaplamak için daha uygundur. F_1 Mikro-ortalama (F_{1mic}) metriğini hesaplamak için kullanılan formüller aşağıdaki gibidir (3.32)-(3.34).

$$P_{mic} = \frac{\sum_{n=1}^N TP_n}{\sum_{n=1}^N TP_n + FP_n} \quad (3.32)$$

$$R_{mic} = \frac{\sum_{n=1}^N TP_n}{\sum_{n=1}^N TP_n + FN_n} \quad (3.33)$$

$$F_{1mic} = 2 \frac{P_{mic} \times R_{mic}}{P_{mic} + R_{mic}} \quad (3.34)$$

Makro-ortalama ve Mikro-ortalama skorlarının birbirine yakın olması beklenmektedir. Mikro-ortalamanın Makro-ortalamadan daha yüksek olması durumunda sınıflandırma yönteminin küçük sınıflar için başarı gösteremediği söylenebilir. Aynı bakış açısıyla Makro-ortalama Mikro-ortalamanın çok üstünde ise, sınıflandırma yönteminin büyük sınıflarda başarı gösteremediği anlaşılmaktadır.

3.1.4. Deney Sonuçları

Bu çalışmanın amacı genel doğruluk değerini koruyarak sınıf içi performansı arttırmaktır. Karşılaştırılan yöntemlerin genel doğruluk değerleri birbirlerine oldukça yakındır.

Dolayısıyla önerilen yöntemler ile ArgMax yönteminin genel doğruluk değerleri hemen hemen aynı değerlere sahiptir (Çizelge 3.6).

Çizelge 3.6 Yaklaşımların genel doğruluk değerleriyle karşılaştırılması

	ArgMax	POEV	NOEV	MOEV
TREMO	90.0	90.0	90.0	90.0
TURTED	75.0	75.0	75.0	75.0
TEC	<u>58.2</u>	58.0	57.5	58.1
Tales	54.4	53.9	54.2	<u>54.7</u>
Daily Dialog	<u>84.2</u>	83.9	83.7	84.1
CrowdFlower	<u>52.0</u>	46.8	51.2	50.7
Blogs	<u>78.6</u>	<u>78.6</u>	75.4	78.3

Önerilen yaklaşımların performansları Çizelge 3.8’te P , R ve F_1 metrikleriyle karşılaştırılmıştır. Tüm sınıflar için önerilen yaklaşımın P değerleri ArgMax yöntemine göre daha düşüktür. Öte yandan R değerleri ise oldukça yüksektir. Önerilen yöntemle TP ve FP değerleri, dolaylı olarak R ve F_1 metriklerinde iyileşme sağlanmıştır. Bu durum özellikle belirsizliğin yüksek olduğu küçük sınıfların sınıflama performansında artış sağlamıştır.

Çizelge 3.7 Türkçe Optimum his vektörleri ve ArgMax yöntemi karşılaştırması

Veri Kümesi	His	ArgMax			POEV			NOEV			MOEV		
		P	R	F ₁	P	R	F ₁	P	R	F ₁	P	R	F ₁
TREMO	<i>öfke</i>	87	88	88	87	88	88	87	88	88	87	88	88
	<i>tiksinme</i>	92	93	93	92	93	93	<u>93</u>	93	93	<u>93</u>	93	93
	<i>korku</i>	91	90	90	91	90	90	<u>92</u>	89	<u>91</u>	<u>92</u>	90	<u>91</u>
	<i>üzüntü</i>	88	89	88	88	89	88	88	89	88	88	89	88
	<i>şaşıрма</i>	92	84	87	92	84	87	<u>93</u>	83	<u>88</u>	92	83	87
	<i>neşe</i>	89	93	91	89	93	91	88	93	91	89	93	91
TURTED	<i>öfke</i>	75	78	77	<u>77</u>	77	77	74	<u>80</u>	77	76	78	77
	<i>tiksinme</i>	77	79	78	77	79	78	76	<u>80</u>	78	77	79	78
	<i>korku</i>	72	74	73	72	74	73	72	73	73	72	73	73
	<i>üzüntü</i>	73	74	73	73	74	73	<u>74</u>	73	73	73	74	73
	<i>şaşıрма</i>	77	77	77	76	<u>78</u>	77	<u>78</u>	75	77	76	77	77
	<i>neşe</i>	80	69	74	79	<u>70</u>	74	<u>81</u>	68	74	80	<u>70</u>	74

Çizelge 3.8 İngilizce Optimum his vektörleri ve ArgMax yöntemi karşılaştırması

Veri Kümesi	His	ArgMax			POEV			NOEV			MOEV		
		P	R	F ₁	P	R	F ₁	P	R	F ₁	P	R	F ₁
TEC	<i>öfke</i>	55	29	38	43	35	38	62	20	31	45	34	<u>39</u>
	<i>tiksinme</i>	60	17	26	36	36	<u>36</u>	57	03	05	46	28	35
	<i>korku</i>	56	51	53	57	52	<u>54</u>	50	54	52	56	52	<u>54</u>
	<i>üzüntü</i>	48	49	48	46	51	48	46	50	48	46	50	48
	<i>şaşıрма</i>	56	46	50	53	51	<u>52</u>	52	49	51	53	50	51
	<i>neşe</i>	64	80	71	70	73	<u>72</u>	66	78	<u>72</u>	69	75	<u>72</u>
Tales	<i>öfke</i>	48	49	<u>48</u>	45	49	47	44	51	47	46	49	<u>48</u>
	<i>tiksinme</i>	44	27	34	39	33	<u>36</u>	44	14	22	41	31	35
	<i>korku</i>	65	45	53	60	48	<u>54</u>	71	41	52	62	47	<u>54</u>
	<i>üzüntü</i>	60	50	54	61	50	55	55	53	54	61	51	<u>56</u>
	<i>şaşıрма</i>	36	30	33	35	33	34	40	32	<u>36</u>	37	33	35
	<i>neşe</i>	58	82	68	62	76	68	59	83	<u>69</u>	61	79	<u>69</u>
Daily Dialog	<i>öfke</i>	46	32	38	41	37	<u>39</u>	43	27	33	42	35	38
	<i>tiksinme</i>	46	18	25	43	27	<u>33</u>	0	0	0	45	24	32
	<i>korku</i>	36	14	20	36	14	<u>20</u>	0	0	0	36	14	<u>20</u>
	<i>üzüntü</i>	67	62	<u>65</u>	66	62	64	61	66	63	66	63	64
	<i>şaşıрма</i>	73	59	65	68	63	<u>66</u>	63	64	64	70	63	<u>66</u>
	<i>neşe</i>	89	96	93	91	95	93	90	96	93	91	95	93
CrowdFlower	<i>öfke</i>	45	23	30	34	42	<u>37</u>	50	03	05	36	38	<u>37</u>
	<i>tiksinme</i>	0	0	0	02	05	<u>03</u>	0	0	0	0	0	0
	<i>korku</i>	45	60	<u>51</u>	49	39	44	45	55	50	48	47	47
	<i>üzüntü</i>	42	23	30	40	36	<u>38</u>	38	26	31	41	34	37
	<i>şaşıрма</i>	0	0	0	12	21	<u>15</u>	0	0	0	13	07	09
	<i>neşe</i>	61	79	<u>69</u>	68	67	68	59	83	<u>69</u>	63	77	<u>69</u>
Blogs	<i>öfke</i>	78	74	76	78	74	76	68	74	71	76	74	75
	<i>tiksinme</i>	78	68	72	76	68	71	76	59	67	75	65	70
	<i>korku</i>	1.0	66	79	1.0	66	79	95	62	75	1.0	66	<u>79</u>
	<i>üzüntü</i>	78	72	<u>75</u>	79	69	74	74	67	70	79	69	74
	<i>şaşıрма</i>	75	48	59	68	60	<u>64</u>	1.0	32	48	70	56	62
	<i>neşe</i>	77	97	86	78	95	86	74	98	84	78	96	<u>86</u>

Önerilen optimum his vektörü yaklaşımları ArgMax yaklaşımı ile karşılaştırıldığında bazı his sınıflarında *R* ve *F₁* performansları oldukça yüksek olduğu görülmektedir (Çizelge 3.8). Örneğin CrowdFlower veri kümesi içerisinde yer alan ve nispeten daha az örneği bulunan “tiksinme” ve “şaşıрма” gibi his sınıflarında ArgMax yöntemi hiç doğru tahminde bulunamazken, optimum his vektörü yaklaşımları için bu olumsuzluk söz

konusu değildir. Çizelge 3.8 içerisinde yer alan 5 veri kümesi ve 30 sınıf içi F_1 performansları değerlendirildiğinde 24 sınıfta önerilen yaklaşımların daha başarılı olduğu görülmektedir. POEV yaklaşımı 7 sınıfta oldukça yüksek olmak üzere 14 sınıfta ArgMax yönteminden daha başarılı olmuştur. POEV yöntemi R ve F_1 metriklerinde oldukça üstündür. Bu sayede önerilen yöntemin sınıf içi başarısı daha yüksektir. POEV ve MOEV kadar başarılı olmasa da NOEV yaklaşımı da POEV yaklaşımına yakın başarı elde etmiştir.

Çizelge 3.9 Optimum His Vektörleri ve ArgMax ortalama metriklerle karşılaştırma

Veri Kümesi	Ortalama	ArgMax			POEV			NOEV			MOEV		
		P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
TREMO	mikro	90	90	90	90	90	90	90	90	90	90	90	90
	makro	90	89	90	90	89	90	90	89	90	90	89	90
TURTED	mikro	75	75	75	75	75	75	75	75	75	75	75	75
	makro	76	75	75	76	75	75	76	75	75	76	75	75
TEC	mikro	58	58	58	58	58	58	58	58	58	58	58	58
	makro	56	45	48	51	50	<u>50</u>	56	42	43	52	48	<u>50</u>
Tales	mikro	54	54	54	54	54	54	54	54	54	55	55	<u>55</u>
	makro	52	47	48	50	48	<u>49</u>	52	46	47	51	48	<u>49</u>
Daily Dialog	mikro	84	84	84	84	84	84	84	84	84	84	84	84
	makro	60	47	51	58	50	<u>53</u>	43	42	42	58	49	52
CrowdFlower	mikro	52	52	<u>52</u>	47	47	47	51	51	51	51	51	51
	makro	32	31	30	34	35	<u>34</u>	32	28	26	33	34	33
Blogs	mikro	79	79	79	79	79	79	75	75	75	78	78	78
	makro	81	71	74	80	72	<u>75</u>	81	65	69	80	71	74

Mikro-ortalama ve Makro-ortalama, bu çalışmadaki gibi dengesiz veri kümelerinde sınıflama yöntemlerinin değerlendirilmesi için önemli göstergelerdir. Makro-ortalama, çok sayıda örneği olan sınıfların performans üzerindeki etkisini göstermektedir. Çizelge 3.9'da yer alan POEV yöntemine ait F_{1mac} sonuçları incelendiğinde, Anma (R) değerlerindeki artıştan kaynaklı olarak, tüm veri kümelerinde ArgMax yönteminden üstündür. Bu durum özellikle daha az örneği olan sınıfların sınıf içi performanslarını iyileştirme iddiasını doğrulamaktadır. Öte yandan Kesinlik (P) metriğinden kaynaklı olarak F_{1mic} değerlerinde bir artış gözlemlenmemiştir. CrowdFlower veri kümesi haricinde POEV yaklaşımının Mikro-ortalama skorları ArgMax ile aynıdır. Öte yandan

POEV yaklaşımının mikro ve makro ortalamaları arasındaki farklar ArgMax yönteminden daha azdır. Bu durum POEV'in daha küçük sınıfları sınıflandırmada daha başarılı olduğu şeklinde yorumlanabilir. NOEV ve MOEV skorları da POEV'in performansını doğrular niteliktedir.

Çizelge 3.10 CrowdFlower veri kümesi POEV ve ArgMax Anma (R) karşılaştırması

His		ArgMax							POEV								
		<i>öfke</i>	<i>tiksinme</i>	<i>korku</i>	<i>üzüntü</i>	<i>şaşıрма</i>	<i>neşe</i>	Anma (R)%	<i>öfke</i>	<i>tiksinme</i>	<i>korku</i>	<i>üzüntü</i>	<i>şaşıрма</i>	<i>neşe</i>	Anma (R) %		
Gerçek Etiket	<i>öfke</i>	64	0	121	44	0	48	23	116	6	55	44	32	24	42		
	<i>tiksinme</i>	5	0	23	5	0	9	0	9	2	14	8	4	5	5		
	<i>korku</i>	30	0	972	192	1	434	60	89	28	642	363	236	271	39		
	<i>üzüntü</i>	26	0	555	236	0	189	23	94	26	285	361	137	103	36		
	<i>şaşıрма</i>	3	0	180	28	0	244	00	13	8	115	48	95	176	21		
	<i>neşe</i>	15	0	316	51	1	1460	79	23	17	197	82	280	1244	67		
Ortalama									31								35

Çizelge 3.10'de CrowdFlower veri kümesi için Anma (R) ve TP skorları ayrıntılı şekilde sunulmaktadır. ArgMax yöntemi “tiksinme ve şaşırma” sınıflarında hiç doğru tahminde bulunamadığı için TP değerleri 0’dır. Bu nedenle bahsedilen sınıflar için Anma değerleri de 0’dır. Öte yandan “şaşıрма” sınıfı için POEV yaklaşımından elde edilen TP puanı 95’tir. Sonuç olarak, sınıf içi Anma performansı “şaşıрма” sınıfı için 0’dan 0.21’e yükseltilmiştir. ArgMax yönteminin ortalama Anma değeri ancak 0.31 seviyesinde kalırken POEV yaklaşımı 0.35 ortalama Anma değeri elde etmiştir.

Çizelge 3.11 TURTED veri kümesi POEV ve ArgMax doğruluk tablosu

His		ArgMax							POEV						
		<i>öfke</i>	<i>tiksinme</i>	<i>korku</i>	<i>üzüntü</i>	<i>şaşıрма</i>	<i>neşe</i>	<i>öfke</i>	<i>tiksinme</i>	<i>korku</i>	<i>üzüntü</i>	<i>şaşıрма</i>	<i>neşe</i>		
Gerçek Etiket	<i>öfke</i>	8636	499	532	815	290	268	8527	507	541	862	303	300		
	<i>tiksinme</i>	510	6471	315	480	162	227	465	6489	316	490	165	240		
	<i>korku</i>	635	441	6009	525	301	264	585	437	6015	538	321	279		
	<i>üzüntü</i>	767	425	554	6324	291	224	716	430	551	6358	297	233		
	<i>şaşıрма</i>	351	160	413	218	4587	217	317	154	406	228	4617	224		
	<i>neşe</i>	571	379	539	283	363	4816	530	373	528	291	366	4863		

Bu tezin ana amaçlarından birisi dengesiz his veri kümeleri üzerinde Türkçe daha başarılı his sınıflaması yapmaktır. Optimum his vektörleri yaklaşımı ile küçük sınıflarda bile başarı gösteren bir yaklaşım ortaya konmuştur. Yöntem küçük sınıflarda başarıyı arttırırken büyük sınıflardaki başarıyı da olumsuz etkilemektedir. Bu yönüyle tezin ana amaçlarından birisi gerçekleştirilmiştir. POEV yaklaşımının ArgMax yöntemiyle karşılaştırıldığı Çizelge 3.9, Çizelge 3.10 ve Çizelge 3.11 bu iddiayı desteklemektedir.

Çizelge 3.12 POEV ve Bostan yaklaşımları performans karşılaştırması

His	Yaklaşım	Blogs	CrowdFlower	Daily Dialogs	Tales	TEC
öfke	Bostan	41	24	18	31	40
	POEV	76	37	39	47	38
tiksinme	Bostan	60	9	13	18	26
	POEV	71	3	33	36	36
korku	Bostan	53	31	16	27	57
	POEV	79	44	20	54	54
üzüntü	Bostan	36	27	20	26	45
	POEV	74	38	64	55	48
şaşıрма	Bostan	25	9	24	23	52
	POEV	64	15	66	34	52
neşe	Bostan	67	38	51	34	68
	POEV	86	68	93	68	72
Mikro Ort	Bostan	76	32	71	54	56
	POEV	79	47	84	54	58

Elde edilen sonuçlar alan yazında bulunan diğer araştırmacılar tarafından raporlanan sonuçlarla karşılaştırılmıştır. Bostan'ın [228] çalışması beş veri kümesini kapsayan ve sınıflandırma sonuçlarını paylaşan çalışmalardan biridir. Bostan çalışmasında *Maksimum Entropi* yöntemini kullanmış ve veri kümeleri üzerinde his sınıflaması yapmıştır. Önerilen POEV yaklaşımıyla elde edilen sonuçlar Bostan'ın raporladığı F_1 ve F_{1mic} sonuçlarıyla karşılaştırılmıştır. F_1 sonuçları karşılaştırıldığında POEV yaklaşımının neredeyse tüm his sınıfları ve tüm his veri kümelerinde üstün olduğu görülmektedir. Aynı şekilde F_{1mic} sonuçlarına bakıldığında ise Tales veri kümesi sonuçlarının aynı, TEC ve Blogs veri kümelerinde ise artış sağlandığı görülmektedir. Özellikle sınıf dağılımının dengesiz olduğu Daily Dialogs ve Crowd Flower veri kümelerinde ise çok belirgin bir performans artışı sağlandığı görülmektedir. Bostan kendi yöntemiyle Crowd Flower veri kümesi üzerinde 0.32 F_{1mic} başarı sağlarken POEV yaklaşımıysa başarıyı %46 artırarak

0.47 F_{mic} skoru elde etmiştir. Bu sonuçlar tezde önerilen yaklaşımın küçük sınıflarda ArgMax yönteminden daha başarılı olmasının yanı sıra başka yöntemlerle de rekabetçi olduğunu göstermektedir. Ayrıca dile özgü hiçbir öznitelik kullanılmadığı için tezde amaçlanan dilden bağımsız bir yöntem olma özelliğini de karşılamaktadır.

Çizelge 3.13 POEV ve ArgMax Belirsizlik giderimi karşılaştırması

Örnek	Etiket	GRU Çıktı Vektörü						Tahmin	
		<i>öfke</i>	<i>tiksinme</i>	<i>korku</i>	<i>üzüntü</i>	<i>şaşıрма</i>	<i>neşe</i>	ArgMax	POEV
yeah i think u deserve a reward for that.....	<i>şaşıрма</i>	0.07894	0.04828	0.16094	0.20125	<u>0.24216</u>	<u>0.26843</u>	neşe	şaşıрма
Breaking dawn..better than i thought	<i>şaşıрма</i>	0.04533	0.03059	0.11969	0.22011	<u>0.27301</u>	<u>0.31127</u>	neşe	şaşıрма
I'm going to have to face it. I suck at statistics and always will.	<i>üzüntü</i>	0.09326	0.04727	0.18207	<u>0.26487</u>	0.11066	<u>0.30186</u>	neşe	üzüntü
I just remembered i need to cancel my hair appt bc i spent my money on other things....	<i>üzüntü</i>	0.08467	0.03718	0.06722	<u>0.32277</u>	0.16137	<u>0.32679</u>	neşe	üzüntü
If this feeling is a precursor to the next month well I just gotta keep my head up.	<i>üzüntü</i>	0.07048	0.02662	0.17449	<u>0.29335</u>	0.12606	<u>0.30900</u>	neşe	üzüntü
Nahi yaar.. has its own benefits...at times makes a person all the more stronger..let it be..Let them work towards it n see	<i>korku</i>	0.13935	0.04316	<u>0.30557</u>	0.10152	0.08617	<u>0.32424</u>	neşe	korku

Önerilen modelin başarısı, TEC veri kümesi test kümesine ait bazı örnekler daha yakından incelenerek gösterilmiştir. Çizelge 3.13’da 6 örnek cümle yer almaktadır. Çizelgede örneklerin etiketleri ve GRU derin öğrenme logit vektöründe yer alan sayısal değerler yer almaktadır. Çizelgenin en sağında ise çıktı vektörüne bakılarak ArgMax yöntemi ve POEV yaklaşımı ile yapılan tahmin sonucunda elde edilen etiketler yer almaktadır. GRU çıktı vektörüne bakıldığında en yüksek üyelik derecesine sahip olan sınıf “neşe” olduğundan ArgMax yöntemi tüm örnekler için “neşe” etiketini tahmin etmiştir. Tabloda yer alan hiçbir örneğin gerçek etiketi “neşe” değildir. GRU çıktılarına dikkatle bakıldığında ikinci en yüksek üyelik derecesiyle en yüksek üyelik derecesi arasındaki farkların oldukça küçük olduğu anlaşılmaktadır. ArgMax yöntemi doğrusaldır

ve aradaki farkın ne ölçüde küçük olduğuna bakmaksızın maksimum üyelik derecesini işaret etmektedir. Öte yandan doğrusal kurallar yardımıyla hangi durumda en yüksek üyelik derecesinin hangi durumda ikinci ya da üçüncü üyelik derecesinin dikkate alınması gerektiğini belirlemek oldukça zordur. Çözüm olarak, bu tezde derin öğrenmenin çıktı vektörüne yerleştirilen, tüm puanları değerlendirmek için POEV yaklaşımı oluşturulmuştur. POEV yaklaşımı sayesinde, çıktı vektörünün puanları arasındaki küçük farklılıklardan kaynaklanan, ArgMax yönteminin verdiği yanlış kararlardan bazıları ortadan kaldırılmıştır.

His analizi üzerine yapılan çalışmaların performansları yaygın olarak genel doğruluk metriğiyle değerlendirilmektedir. Bir sınıflama yönteminin genel doğruluk performansı, her bir sınıfın başarısını göstermeye yetmemektedir. Klasik derin öğrenme yaklaşımı genel doğruluk metriğine odaklanmış olduğundan bazı sınıflar için herhangi bir başarı gösterememektedir. Bu tez kapsamında önerilen yaklaşımla, genel doğruluk skorlarında herhangi bir olumsuz değişiklik olmadan sınıf içi performanslar artırılmaktadır. Sınıflama yönteminin başarısı ölçülürken, özellikle dengesiz dağıtılmış veri kümeleri için, genel doğruluk metriğinin yanı sıra sınıf içi performansın da hesaba katılması gerekmektedir.

Bu çalışmanın temel motivasyonu, his analizinde sadece genel doğruluk metriğiyle performans değerlendirmesinin çok iyimser olması ve sınıf içi performansının adil bir değerlendirme için yeniden gözden geçirilmesi gerektiğidir. Bu bağlamda, derin öğrenme ve sürü zekası optimizasyon yöntemlerini birleştirerek daha iyi performans gösteren bir yaklaşım önerilmektedir. Deneyler, optimize edilmiş vektörlerin derin öğrenme tarafından üretilen ağırlıklı vektör için doğru his etiketinin tahminine rehberlik ettiğini göstermektedir. Bu yaklaşım iyi bilinen beş veri kümesiyle incelenerek genelleştirilebilir bir sonuç olarak doğrulanmıştır. Sonuç olarak, ortaya çıkan sınıf etiketlerinin seçiminde kullanılan doğrusal yaklaşımlardan kaçınılması gerekmektedir. Bu nedenle, doğru ve etkili performans ölçümü için her his sınıfı dikkate alınmalıdır.

4.ÖN EĞİTİMLİ DİL MODELLERİ İLE TÜRKÇE HIS ANALİZİ

Bu çalışmada az örneğin bulunduğu Türkçe his sınıflama bağlamına ön eğitilmiş dil modellerinden elde edilen genel bilgi transfer edilerek, daha başarılı Türkçe his sınıflama gerçekleştirilmektedir. Geleneksel makine öğrenmesi yöntemleri girdi olarak ham metin yerine öznitelikler kullanılmaktadır. Bu öznitelikler metinlerden elde edilen kelime çantası (Bag of Words - BOW) [1], kelime n-gramları [2], noktalama işaretleri, emoji ve his sözlüklerinden elde edilen öznitelikler [3] olmaktadır. Ayrıca girdinin kalitesini arttırmak ve boyutunu azaltmak amacıyla bu öznitelikler üzerinde seçme, ağırlıklandırma ve çıkarma gibi öznitelik mühendisliği işlemleri yapılmaktadır. Gerçekleştirilen bu öznitelik mühendisliği işlemleri metin içerisinde yer alan bazı önemli bilgilerin yok olmasına ve sınıflama başarısının düşmesine sebep olmaktadır. Ayrıca girdilerin temsil edildiği BOW gibi yöntemler cümle içerisinde yer alan kelimelerin sırasının ve bütünlüğünün kaybolmasına sebep olmaktadır.

Geleneksel makine öğrenmesi yöntemlerinde karşılaşılan bu olumsuzlukları ortadan kaldırmak amacıyla derin öğrenme yöntemleri kullanılmaktadır. Derin öğrenme yöntemleri metni ham haliyle girdi olarak almakta ve bir arada geçen kelimelerin bağlamını korumaktadır. Bu sayede geleneksel makine öğrenmesi yöntemlerine göre daha başarılı his sınıflama yapılmasına olanak sağlanmaktadır. Öte yandan derin öğrenme yöntemleri birçok gizli katmanı bulunan katmanlar arasında ağırlıkların her iki yönde güncellendiği bir tür yapay sinir ağlarıdır. Bu katmanlar arası ağırlıkların optimize edilmesiyle hem bağlam belirlenmekte hem de kıymetli öznitelikler ön plana çıkmaktadır. Bu sayede his sınıflama modeli ortaya oluşturulmaktadır. Güncel çalışmalarda geleneksel sınıflama algoritmalarından daha iyi sonuçlar elde etmek için, Convolutional Neural Network (CNN)[90] ve Recurrent Neural Network (RNN)[191] gibi derin öğrenme yöntemleri kullanılmaktadır. Ancak derin öğrenme yöntemlerinde kaliteli bir model ortaya çıkarabilmek için büyük veriye ihtiyaç duyulmaktadır. Türkçe his analizi konusunda etiketlenmiş veri miktarı oldukça azdır. Bu sebeple derin öğrenme yöntemleriyle oluşturulan his sınıflama modellerinin başarısı kısıtlanmaktadır.

İnsan beyni bir bilgiyi öğrenirken daha önce benzer konularda öğrendiği bilgileri taban olarak kullanmaktadır. Bu özelliği taklit etmek amacıyla genel amaçlı ön eğitilmiş kelime

özyerleşikleri [4–6] üretilmiş ve bazı İngilizce çalışmalarda [90] his analizine adapte edilerek kullanılmıştır. Öte yandan kelime seviyesinde ön bilgi sağlayan bu yaklaşımın yanında cümle seviyesinde çalışan yaklaşım [7] da mevcuttur. Kelime anlamını yakalamayı amaçlayan tüm bu gelişmeler genel amaçlı ön eğitilmiş dil modeli [9] yaklaşımını tetiklemiştir. Ön eğitilmiş dil modelleri metin içerisinde yer alan kelime anlamlarının ötesinde semantik, sentaktik ve dil bilgisi kurallarının yanı sıra kullanım alışkanlıklarını öğrenmeyi hedeflemektedir. Çok büyük veri üzerinden multi-GPU süper bilgisayarlar yardımıyla öğrenilen evrensel dil modeli genel amaçlıdır ve birçok alt görevde kullanılabilir. Son yıllarda ön eğitilmiş dil modeli yaklaşımlarında yaşanan gelişmeler doğal dil araştırmalarının yönünü değiştirmiştir. Genel amaçlı bir dil modeli oluşturmak ve alt problemlere uyarlayarak kullanmak bir endüstri standardı haline gelmektedir. Derin ağlarda ön eğitilmiş dil modeli kullanımı sayesinde eğitim sıfırdan başlamaz bu sayede modelin genelleşmesi ve problemi kavraması hızlanmaktadır. Ön-eğitilmiş dil modelleri sayesinde derin öğrenme yöntemlerinde ihtiyaç duyulan büyük miktardaki his verisi ihtiyacı ortadan kalkmaktadır.

İngilizce için yaşanan bu gelişmeler Türkçe ön eğitilmiş dil modellerinin oluşturulmasını tetiklemiştir. Türkçenin eklemeli bir dil olması ve ekler yardımıyla yeni kelime üretme yeteneğinin olması sebebiyle Türkçe morfolojik açıdan zengin diller sınıfında yer almaktadır. Türkçe ön eğitilmiş modeller oluşturmak için İngilizcedekinden daha fazla örneğe ihtiyaç duyulmaktadır. Bu sebeple Türkçe dil modeli eğitilirken kelime hazinesini yüksek tutmak için erişilebilen tüm yazılı kaynaklar kullanılmaktadır. Literatürde bu yaklaşım ile Türkçe[12] ve Türkçenin de aralarında olduğu çok dilli[229] ön eğitilmiş dil modelleri oluşturulmuştur. Bu çalışmada ilk kez Devlin ve ekibi tarafından geliştirilen çok dilli “BERT” (Bidirectional Encoder Representations from Transformers) [9] ve Türkçe için geliştirilen BERTurk[12] ön eğitilmiş dil modelleri his analizi problemine adapte edilmiş ve ilgili veri kümeleri üzerinde kısa sürede yüksek başarı elde edilmiştir. Buna ek olarak konvansiyonel makine öğrenmesi ve derin öğrenme yöntemleri ile aynı deneyler tekrar edilmiş ve ayrıntılı biçimde tartışılmıştır.

4.1.1. Yöntem

Bu çalışmada, Devlin ve arkadaşları tarafından geliştirilmiş olan çok dilli BERT [9] ön eğitilmiş genel amaçlı dil modeli kullanılmıştır. Devlin ve ekibi BERT modeli için sayısal ortamda yer alan Wikipedia gibi metin kaynaklarının yanı sıra var olan veri kümelerini dahil ederek 3,3 milyar kelimelik büyük bir veri kaynağı oluşturmuştur. Veri kaynağından alınan metin; cümle sınırlarını, kelimeleri ve kelime konumlarını temsil eden BERT gösterimine dönüştürülerek Transformer algoritmasına [143] girdi olarak vermişlerdir. BERT gösteriminde her girdi sınıflamaya özel anahtar [CLS] ile başlar ve cümleler birbirlerinden ayraç anahtarı [SEP] ile ayrılır. Girdi içerisindeki kelimelerin %15'i seçilmiş ve [MASKE] özel anahtarıyla değiştirilerek BERT modelinin bu maskelenmiş kelimeleri öğrenmesi sağlanmıştır. Bu ön işlemlere tabi tutulan girdiler maskelenmiş girdiler olarak anılmaktadır. Modelin sadece maskeleri öğrenmek yerine tüm terimleri ve eş dizimleri öğrenmesi amacıyla, maskelenen kelimenin rastgele kelimelerle değiştirilmesi ve hiç dokunulmadan eğitime girdi olarak verilmesi gibi önlemler alınmıştır. Öte yandan sonraki cümleyi tahmin etmek amacıyla ikinci cümlelerin %50'si rastgele seçilmiş cümleler ile değiştirilerek algoritmaya negatif örnek olarak verilmiştir. Derin ağ 24 Transformer bloğu, 1024 gizli katman ve 340 milyon parametre ile eğitilmiştir. Farklı blok büyüklükleriyle İngilizce ve diğer diller için üretilmiş ön eğitilmiş dil modelleri farklı problemlerde kullanılması amacıyla erişime açılmıştır¹. Parametrelerin damıtılarak azaltıldığı ve model boyutunun küçültüldüğü DistilBert[144] gibi versiyonları da mevcuttur.

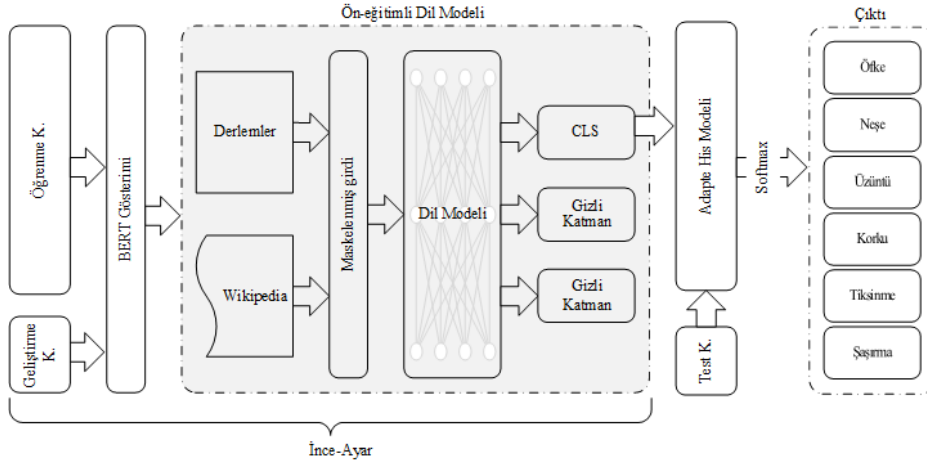
Transfer Learning bir alanda veya görevde edinilen bilginin başka bir alana veya göreve adapte edilmesi işlemine verilen isimdir. Bilgi transferi için genellikle kullanılan iki yöntem vardır. İlk yöntemde öznitelik çıkarımı yapmak amacıyla ön-eğitilmiş parametreler dondurulmuştur[5,106]. İkinci yöntemde ise ön-eğitilmiş parametreler dahil olmak üzere ağdaki tüm parametreler İnce-Ayar (*Fine-Tune*) yapılmaktadır[179]. İlk yöntem daha komplike işler için kullanılmaktayken ikincisi ise daha genel kullanıma yöneliktir. İnce-ayar yapmak amacıyla ön-eğitilmiş modelin üst katmanı alınabileceği gibi tüm katmanlar alınıp uygun katman seçimi modele de yaptırılabilir[160].

¹ <https://huggingface.co/models>

Bu çalışmada BERT mimarisindeki ön-eğitilmiş dil modelleri içerisindeki bilgi, his sınıflama işi için ince-ayar yapılmaktadır. Bu işlem için BERT gösteriminde sınıflandırma [CLS] özel anahtarı kullanılmaktadır. [CLS] BERT gösteriminde ilk anahtardır ve içerisinde tüm sekans temsil edilmektedir. İnce-ayar işlemi için ön-eğitilmiş modellerin son gizli durumlarında h yer alan [CLS] anahtarı sonrasında c sınıfının ihtimalini tahmin etmek amacıyla bir Softmax katmanı eklenmiştir (4.35).

$$p(c|s) = \text{softmax}(W \cdot h + b) \quad (4.35)$$

Denklemden yer alan W ağırlık matrisi b ise *Bias*'tır. Bu sayede ön eğitilmiş dil modellerinden edinilen ağırlıklar his sınıflama bağlamına göre yeniden düzenlenmektedir. Softmax fonksiyonu içine aldığı vektör elemanlarının 0-1 aralığında olmasını ve toplamalarının bir olmasını garanti etmektedir.



Şekil 4.18 Önerilen ön eğitilmiş dil modelleri ile Türkçe his analizi yöntemi

Şekil 4.18'de görüleceği üzere his veri kümesi öğrenme kümesi ön eğitilmiş dil modeline girdi olarak verilmektedir. Öğrenme kümesi içerisindeki örneklere göre ön eğitilmiş dil modeli ağırlıkları güncellenmektedir. Geliştirme kümesi üzerinde başarı gösteren ince-ayar yapılmış model ile test veri kümesi denenmektedir.

Bu çalışma kapsamında his analizine adapte etmek üzere Multilingual BERT[9] ve BERTurk[12] Türkçe ön eğitilmiş dil modelleri kullanılmıştır. Oluşturulan çerçevenin doğrulaması amacıyla İngilizce his sınıflama kümeleri için BERT[9], DistilBert[230] İngilizce ön eğitilmiş dil modelleri de kullanılmıştır. Önerilen yöntemde ince-ayar işlemi için SimpleTransformer² kütüphanesi kullanılmıştır. Kütüphane Transformers[231] altyapısını kullanmaktadır. İnce-ayar işlemi esnasında sistem parametreleri “batch-size:8”, ”learnin-rate:4e-5”, “num-train-epoch:5” şeklinde kullanılmıştır. Hesaplama işlemleri Colab³ ortamında GPU üzerinde gerçekleştirilmiştir. Devamında eğitilen modeller test kümeleri üzerinde denenmiş ve değerlendirme metrikleri ile sonuçlar raporlanmıştır.

4.1.2. His Veri Kümeleri

Çalışma kapsamında Türkçe his sınıflandırma için elle oluşturulmuş ve doğrulanmış olan TREMO[32] ve Türkçe twitter gönderilerinden oluşan TURTED[191] Türkçe his veri kümesi kullanılmıştır. Veri kümeleri deneylerde kullanım için “öğrenme, geliştirme ve test” olmak üzere üç bölüme ayrılmıştır (Çizelge 3.5).

TREMO Türkçe his veri kümesi [32] Ekman temel hisleri ile etiketlenmiş ve dengeli bir dağılıma sahiptir. Veri kümesi 4.709 kişi üzerinde yapılan anket sonucunda oluşturulmuştur. Anket kapsamında gönüllülere her bir Ekman hissi için bir anı ya da deneyimini kısaca yazması istenmiştir. Elde edilen 27.350 girdi doğrulama için en az üç farklı kişi tarafından etiketlenmiştir. Etiketleme neticesinde oy birliği ve oy çokluğu oluşmayan girdiler elenmiştir. Bu çalışmada eleme sonrasında kalan 25989 cümle ile deneyler gerçekleştirilmiştir (Çizelge 3.5).

TURTED Türkçe his veri kümesi [191] Ekman temel hisleri ile etiketlenmiş Türkçe twitter gönderilerinden oluşmaktadır. Twitter gönderilerinde TREMO veri kümesi içerisinde seçici özelliği yüksek kelimeler aranarak elde edilmiş veri kümesi içerisinde

² <https://simpletransformers.ai/>

³ <https://colab.research.google.com/>

195.445 gönderi bulunmaktadır. Gönderiler Twitter'a özgü bilgilerden, sayılardan, gülen yüz ve his ikonlarından temizlendikten sonra deneylerde kullanılmıştır (Çizelge 3.5).

4.1.3. Deneysel Çalışmalar

Deneyler kapsamında öncelikle geleneksel makine öğrenmesi yöntemleri devamında derin öğrenme yöntemleri, kelime özyerleşikleriyle birlikte derin öğrenme yöntemleri ve son olarak ön eğitilmiş dil modelleri ince-ayar yapılarak his sınıflandırma yapılmıştır. Ayrıca Türkçe girdilerde kelime yerine kelime gövdesi almanın etkisini ölçmek amacıyla yapılan tüm deneyler hem tam kelime hem de gövde olarak tekrarlanmıştır. Gövdeleme için Türkçe gövdeleyici⁴ kullanılmıştır.

Öncelikle bir referans noktası oluşturma ve alan yazın ile karşılaştırma [32] amacıyla Naive Bayes (NB)[50] ve SVM[49] geleneksel makine öğrenmesi yöntemleri ile his sınıflama yapılmıştır. Konvansiyonel yöntemlere girdi olarak ham metin girilemediğinden öznitelik oluşturma, ağırlığa göre seçme işlemleri gerçekleştirilmiştir. Öğrenme kümesi içerisinde kelimeler ve n-gramlar(2,3,4) arasından TF-IDF ağırlıklarına göre temsil yeteneği en yüksek 4000 öznitelik seçilmiş ve deneylere girdi olarak kullanılmıştır.

Devamında derin öğrenme algoritmalarından GRU[99], LSTM[90] ve Attention[232] ağlarını çift yönlü kullanarak his sınıflama yapılmıştır. Derin ağların *Embedding* katmanına *FastText*[6] ön eğitilmiş kelime özyerleşikleri ağırlıkları eklenerek GRU, LSTM ve Attention yöntemleriyle deneyler tekrarlanmıştır. Bu deneylere girdi olarak ham metin kullanılmıştır. 30 epoch boyunca eğitilen modellerde aktivasyon fonksiyonu olarak Softmax[233], kayıp fonksiyonu olarak “*Kullback Leiber Divergence*” [234] ve optimizasyon yöntemi olarak “*Nadam*”[235] kullanılmıştır.

⁴ <https://github.com/otuncelli/turkish-stemmer-python>

4.1.4. Performans metrikleri

Diğer çalışmada kullanılan ve Bölüm 3.1.3 te tarif edilen Kesinlik (Precision – P), Anma (Recall – R), F_1 ve Genel Doğruluk (Accuracy – Acc) performans metrikleri kullanılmaktadır.

4.1.5. Deney Sonuçları

Önerilen ön eğitilmiş dil modeli adapte edilmiş his sınıflama modeliyle gerçekleştirilen deneyler sonucunda TREMO veri kümesi üzerinde %92, TURTED veri kümesi üzerinde %79 doğruluk başarısı elde edilmiştir. Bu skorlar alan yazındaki en yüksek sonuçlardır.

Çizelge 4.14 BERTurk[12] ön-eğitilmiş dil modeliyle Türkçe His Sınıflama Sonuçları

His	TREMO									TURTED								
	Öğrenme			Geliştirme			Test			Öğrenme			Geliştirme			Test		
	P	R	F ₁	P	R	F ₁	P	R	F ₁	P	R	F ₁	P	R	F ₁	P	R	F ₁
Öfke	100	100	100	92	93	93	90	92	91	95	95	95	80	80	80	79	81	80
Tiksinme	100	100	100	94	95	94	95	95	95	95	95	95	81	79	80	81	79	80
Korku	99	99	99	90	93	92	91	94	92	92	95	94	77	78	78	76	78	77
Neşe	100	100	100	94	93	94	95	95	95	96	94	95	79	80	79	80	78	79
Üzüntü	100	100	100	92	92	92	92	90	91	94	92	93	76	75	76	78	76	77
Şaşırma	100	99	100	92	88	90	91	87	89	95	96	95	79	81	80	80	81	80
Doğruluk			100			92			92			95			79			79
Makro ort	100	100	100	92	92	92	92	92	92	95	95	95	79	79	79	79	79	79
Ağırlıklı ort	100	100	100	92	92	92	92	92	92	95	95	95	79	79	79	79	79	79

Çizelge 4.14 de görüldüğü üzere TREMO veri kümesi içerisinde en başarısız sınıf olan “şaşırma” sınıfında %89, TURTED veri kümesi içerisinde en başarısız sınıflar olan “korku ve üzüntü” sınıflarında %77 başarı sağlanmıştır. Öte yandan TREMO veri kümesinde “tiksinme” ve “neşe” sınıflarında %95, TURTED veri kümesi “öfke, tiksinme ve şaşırma” sınıflarında %80 başarı elde edilmiştir. Makro ve ağırlıklı ortalama sonuçlarına bakıldığında tüm sınıflarda elde edilen başarının benzer olduğu görülmektedir. Buda modelin başarımının sınıflara ait örnek sayılarından bağımsız olduğunu göstermektedir.

Çizelge 4.15 TREMO Test kümesi karmaşıklık matrisi

His		Öfke	Tiksinme	Korku	Neşe	Üzüntü	Şaşırma	Toplam	R%
Gerçek Etiket	Öfke	1089	23	12	6	40	11	1181	92
	Tiksinme	19	856	20	0	5	5	905	95
	Korku	19	11	1027	6	28	7	1098	94
	Neşe	13	3	11	1237	16	28	1308	95
	Üzüntü	43	8	41	14	1133	16	1255	90
	Şaşırma	23	4	12	44	14	654	751	87
Tahmin edilen Etiket								6498	92

Çizelge 4.15 içerisinde TREMO veri kümesi Test setinde yer alan tüm sınıflara ait örnek sayıları ve tahmin edilen etiketlerin yer aldığı karmaşıklık matrisi (*Confusion Matrix*) görülmektedir. Tahmin edilen etiket sayılarının oldukça başarılı olduğu görülmektedir. Tahmin başarısının en düşük olduğu sınıf olan “şaşırma” sınıfının diğer sınıflara göre daha az örneğe sahip olduğu görülmektedir. Train ve dev örnek sayılarının (Çizelge 3.5) da benzer dağılım gösterdiği hesaba katıldığında başarının görece düşük olması örnek sayısı ile doğru orantılıdır.

Çizelge 4.16 TURTED Test kümesi karmaşıklık matrisi

His		Öfke	Tiksinme	Korku	Neşe	Üzüntü	Şaşırma	Toplam	R
Gerçek Etiket	Öfke	8895	480	484	298	637	246	11040	81
	Tiksinme	504	6470	336	299	388	168	8165	79
	Korku	408	299	6413	311	437	307	8175	78
	Neşe	400	209	436	5407	229	270	6951	78
	Üzüntü	729	437	468	199	6512	240	8585	76
	Şaşırma	261	122	331	234	195	4803	5946	81
Tahmin edilen Etiket								48862	79

Çizelge 4.16 içerisinde TURTED veri kümesi Test setinde yer alan tüm sınıflara ait örnek sayıları ve tahmin edilen etiketlerin yer aldığı karmaşıklık matrisi (*Confusion Matrix*) görülmektedir. Tahmin edilen etiket sayılarının oldukça başarılı olduğu görülmektedir. En küçük sınıf olan “şaşırma” sınıfına ait örnek sayısının 5946 olduğu görülmektedir. Tüm sınıflarda modelin öğrenmesine yetecek miktarda veri olduğu için sınıf büyüklükleri TURTED veri kümesi üzerinde yapılan his analizi sonuçlarını olumsuz olarak etkilememiştir.

Çizelge 4.17 Farklı ön-eğitilmiş dil modelleriyle Türkçe his analizi sonuçları

Ön eğitilmiş dil modeli	TREMO			TURTED		
	P	R	F ₁	P	R	F ₁
BERT(çok dilli)	89	89	89	79	79	79
DistilBERT(çok dilli)	89	88	88	79	79	79
BERTurk	92	92	92	79	79	79
DistilBERTurk	90	91	91	79	79	79

Türkçe his sınıflama modeline uyarlamak üzere birden fazla ön eğitilmiş dil modeli denenmiştir. Çok dilli BERT modeli Wikipedia üzerinde yeterli kaynağa sahip olan tüm diller için geliştirildiğinden Türkçeye özgü kaynak ve yöntemler kullanılmamıştır. Öte yandan ön eğitilmiş dil modelini damıtarak boyutunu düşürmeyi amaçlayan DistilBERT yaklaşımı kaynak olarak BERT modelini kullanmaktadır. Ancak BERTurk ön eğitilmiş dil modelinin geliştirimi esnasında Türkçeye özgü bazı kaynak ve araçlar kullanıldığından [12] Türkçe için diğerlerine göre daha başarılı sonuçlar üretmiştir (Çizelge 4.17). Model büyüklüğü oldukça az olmasına rağmen kaynak olarak BERTurk modelini kullandığı için DistilBERTurk ön eğitilmiş dil modeli de rekabetçi sonuçlar üretmeyi başarmıştır.

Çizelge 4.18 Kelime Özyerleşiklerinin derin öğrenme sonuçlarına etkisi

Method	TREMO						TURTED					
	DL			DL + FastText			DL			DL + FastText		
	P	R	F ₁	P	R	F ₁	P	R	F ₁	P	R	F ₁
GRU	84	83	83	90	89	90	70	70	70	76	75	75
LSTM	85	84	84	90	89	89	68	68	68	75	74	75
Attention	85	85	85	86	86	86	67	66	66	73	73	73

Derin ağların *Embedding* katmanlarında yer alan ağırlıkların sıfırdan başladığı durumda öğrenme başarısı örnek ve *Epoch* sayısına bağlı olarak değişim göstermektedir. Çizelge 4.18'te yer alan sıfırdan başlayan derin öğrenme ve FastText kelime özyerleşikleri ağırlıklarıyla başlayan derin öğrenme sonuçları incelendiğinde, ön eğitilmiş kelime özyerleşiklerinin oldukça güçlü bir ön bilgi sağladığı görülmektedir.

Çizelge 4.19 TREMO Önerilen yöntem ve diğer yöntemlerin sonuçları

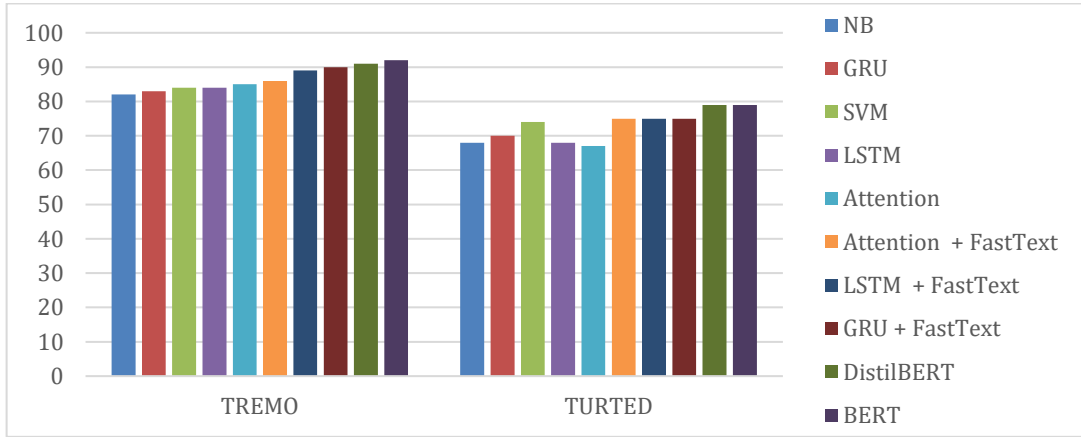
	Raporlanan SVM [32]			SVM-Kelime			SVM-Gövde			Bi-GRU + FastText			BERTurk		
	P	R	F ₁	P	R	F ₁	P	R	F ₁	P	R	F ₁	P	R	F ₁
Öfke	85	87	86	79	85	82	77	84	80	85	90	87	90	92	91
Tiksinme	95	91	93	90	85	87	89	85	87	94	92	93	95	95	95
Korku	89	88	88	89	85	87	87	85	86	91	90	91	91	94	92
Neşe	81	89	85	83	85	84	82	86	84	90	92	91	95	95	95
Üzüntü	85	81	83	76	82	79	78	80	79	88	87	88	92	90	91
Şaşırma	88	82	85	91	76	83	91	76	83	91	85	88	91	87	89
Makro ort	87	86	86	85	83	84	84	83	83	90	89	90	92	92	92
Ağırlıklı ort	87	86	86	84	83	83	83	83	83	90	90	90	92	92	92

Çizelge 4.19'e bakıldığında geliştirilen ön eğitilmiş his modelinin hem alan yazında raporlanan sonuçtan hem de diğer öğrenme yöntemlerinden daha başarılı olduğu görülmektedir. Ayrıca SVM yöntemine girdi olarak kelime gövdeleri (SVM-Gövde) ve kelimenin kendisi verilerek (SVM-Kelime) gövde kullanmanın etkisi incelenmiş ve gövde kullanmanın etkisiz olduğu sonucuna varılmıştır. Ayrıca geleneksel makine öğrenmesi yöntemlerinde ihtiyaç duyulan öznelik mühendisliği işlemleri yerine derin öğrenme yöntemlerine ön bilgi transferinin daha başarılı olduğu görülmektedir. FastText kelime özyerleşikleri kullanılan derin öğrenme yönteminin başarısını %7 arttırdığı görülmektedir. Ön bilgi transferinin önemi bu sonuçlara bakıldığında açıkça görülmektedir.

Çizelge 4.20 TURTED Önerilen yöntem ve diğer yöntemlerin sonuçları

	Raporlanan LSTM [191]			Bi-LSTM			Bi-LSTM +FastText			BERTurk		
	P	R	F ₁	P	R	F ₁	P	R	F ₁	P	R	F ₁
Öfke	70	79	74	68	74	71	75	78	76	79	81	80
Tiksinme	75	73	74	69	73	71	75	79	77	81	79	80
Korku	73	70	72	65	68	67	73	72	72	76	78	77
Neşe	71	69	70	66	66	66	76	70	73	80	78	79
Üzüntü	72	72	72	69	64	67	74	72	73	78	76	77
Şaşırma	76	71	74	75	62	68	75	76	76	80	81	80
Makro ort	73	73	73	68	68	68	75	74	75	79	79	79
Ağırlıklı ort	73	73	73	69	68	68	75	75	75	79	79	79

Çizelge 4.20 incelendiğinde önerilen ön eğitilmiş his modelinin hem alan yazında karşılaşılan sonuçlardan hem de diğer yöntemlerle elde edilen sonuçlardan daha başarılı olduğu görülmektedir. FastText kelime özyerleşiklerinden elde edilen ön bilginin Bi-LSTM sonuçlarını %7 arttırdığı görülmektedir. Ön eğitilmiş dil modelinden elde edilen bilginin ise Bi-LSTM sonuçlarını %11 arttırdığı görülmektedir.



Şekil 4.19 His sınıflama sonuçları

TREMO üzerinde gerçekleştirilen Türkçe his sınıflama için geleneksel makine öğrenmesi yöntemleri en düşük performansı sergilerken, onları derin öğrenme yöntemleri takip etmektedir. Derin öğrenme yöntemlerinde kelime özyerleşiklerinden gelen ön bilgi transfer edildiğinde başarı oldukça artmaktadır. Ön eğitilmiş dil modelleri kullanılarak transfer edilen ön bilgi arttırıldığında ise en yüksek başarı elde edilmektedir. Gerçekleştirilen deneyler sonucunda TURTED his veri kümesi için de bu örüntünün yaklaşık olarak aynı olduğu belirlenmiştir (Şekil 4.19).

Çizelge 4.21 görüldüğü üzere TREMO ve TURTED veri kümeleri deney sonuçlarında Türkçe his sınıflama sonuçlarında benzer bir örüntü elde edilmiştir. Geleneksel makine öğrenmesi yöntemleri en başarısız sonuçları üretirken ön eğitilmiş dil modelleriyse en başarılı sonuçları vermektedir. TREMO veri kümesi elle oluşturulmuş ve doğrulanmış olduğu için sosyal medya gönderilerinden alınarak oluşturulan TURTED veri kümesinden daha başarılı şekilde his sınıflaması yapılmıştır.

Çizelge 4.21 Gerçekleştirilen tüm deney sonuçları

Veri Kümesi	Veri Kümesi Parçası	Metrik	SVM	NB	GRU	GRU + Fasttext	LSTM	LSTM + Fasttext	Attention	Attention + Fasttext	BERTurk	DistilBERTurk
TREMÖ	Öğrenme	P	92	88	99	94	100	93	100	100	100	100
		R	91	87	99	94	99	92	100	100	100	100
		F ₁	91	87	99	94	100	92	100	100	100	100
	Geliştirme	P	85	84	85	90	85	90	86	86	92	91
		R	83	82	83	90	84	89	85	86	92	91
		F ₁	84	83	84	90	84	90	85	86	92	91
	Test	P	85	83	84	90	85	90	85	86	92	90
		R	83	82	83	89	84	89	85	86	92	91
		F ₁	84	82	83	90	84	89	85	86	92	91
TURTED	Öğrenme	P	79	74	99	81	99	81	99	99	95	90
		R	79	69	99	81	99	80	99	99	95	90
		F ₁	79	71	99	81	99	81	99	99	95	90
	Geliştirme	P	74	72	69	76	68	74	67	74	79	79
		R	73	67	69	75	67	74	66	74	79	79
		F ₁	73	68	69	75	67	74	66	74	79	79
	Test	P	74	72	70	76	68	75	67	73	79	79
		R	73	67	70	75	67	74	66	73	79	79
		F ₁	74	68	70	75	68	75	66	73	79	79

Gerçekleştirilen tüm deneyler sonucunda oluşturulan ön eğitilmiş his modeli hem alan yazında alınan sonuçlardan hem diğer tüm yöntemlerden daha başarılı sonuçlar üretmektedir. Bu durum her iki veri kümesi sonuçları için de geçerlidir. His bağlamına özel hiçbir öznelik kullanmaksızın gerçekleştirilen deneyler sonucunda alan yazındaki en yüksek sonuç elde edilmiştir.

4.1.6. Ön Eğitilmiş His Modeli için Optimum His Vektörleri

Tez kapsamında önerilen ilk yöntem olan Optimum His Vektörleri ile derin öğrenme karar mekanizmasında bir değişiklik önerilmiş ve bu değişiklik sayesinde belirsizliğin yüksek olduğu görece küçük sınıfların sınıf içi başarısı artırılmıştır. Bu bölümde önerilen yaklaşımda ise ön eğitilmiş dil modelleri Türkçe his sınıflama problemine ince ayar yapılarak az veri üzerinden daha başarılı bir his modeli oluşturulmuştur. Bu alt bölümde

ise Optimum His Vektörleri ile önerilen karar mekanizması BERTurk ile eğitilen ön eğitilmiş his modeli üzerinde denenerek mekanizmanın katkısı tartışılmıştır.

Ön eğitilmiş his modelinden elde edilen çıktı vektörleri üzerinden POEV, NOEV ve MOEV Optimum his vektörleri elde edilmiş ve standart karar mekanizması olan ArgMax yöntemiyle karşılaştırılmıştır.

Çizelge 4.22 Ön Eğitilmiş His Modeli için Optimum His Vektörleri etkisi

Veri Kümesi	His	ArgMax			POEV			NOEV			MOEV		
		P	R	F ₁	P	R	F ₁	P	R	F ₁	P	R	F ₁
TREMÖ	<i>öfke</i>	92	90	91	92	90	91	92	90	91	92	90	91
	<i>tiksinme</i>	94	94	94	94	94	94	94	94	94	94	94	94
	<i>korku</i>	90	95	92	90	95	92	90	95	<u>93</u>	90	95	<u>92</u>
	<i>neşe</i>	95	94	95	95	94	95	95	94	95	95	94	95
	<i>üzüntü</i>	91	91	91	91	91	91	91	91	<u>91</u>	91	91	91
	<i>şaşıрма</i>	90	88	89	90	88	89	90	88	89	90	88	89
TURTED	<i>öfke</i>	79	81	80	<u>80</u>	80	80	78	<u>82</u>	80	79	81	80
	<i>tiksinme</i>	81	79	80	81	79	80	81	79	80	81	79	80
	<i>korku</i>	76	78	77	76	78	77	76	<u>79</u>	77	76	<u>79</u>	77
	<i>üzüntü</i>	80	78	79	80	78	79	<u>81</u>	77	79	80	78	79
	<i>şaşıрма</i>	78	76	77	77	76	77	78	75	77	78	76	77
	<i>neşe</i>	80	81	80	79	81	80	81	79	80	80	80	80

Gerçekleştirilen deneyler sonucunda Çizelge 4.22 içerisinde görüldüğü üzere TURTED veri kümesi üzerinde daha başarılı sonuçlar elde edilmiştir. Önerilen Optimum His Vektörleri yaklaşımı belirsizliği yüksek olan veri kümelerinde daha başarılı sonuçlar üretmektedir. TREMO veri kümesi elle oluşturulmuş ve doğrulanmış olduğundan belirsizliği oldukça azdır. Bu sebeple TREMO veri kümesi üzerinde NOEV haricinde bir performans artışı gerçekleşmemiştir. Ancak TURTED veri kümesi doğal ortamdaki elde edildiği için belirsizliği yüksektir ve bu nedenle önerilen belirsizlik giderimi yöntemi ile performans artışı sağlanmıştır. Çizelge 4.22 incelendiğinde Optimum His vektörlerinin tüm sınıflarda en az ArgMax yöntemi kadar başarılı olduğu görülmektedir. Öte yandan belirsizliğin yüksek olduğu TURTED veri kümesi içerisinde “öfke, korku ve üzüntü” sınıflarında, belirsizliğin az olduğu TREMO veri kümesi içerisinde “korku ve üzüntü” sınıflarında performans artışı gerçekleştirilmiştir.

Çizelge 4.23 TREMO Ön Eğitimli His Modeli ArgMax ve NOEV doğruluk matrisi

His		ArgMax						NOEV					
		öfke	tiksinme	korku	neşe	üzüntü	şaşıрма	öfke	tiksinme	korku	neşe	üzüntü	şaşıрма
Gerçek Etiket	öfke	1068	22	19	6	54	12	1068	22	18	6	55	12
	tiksinme	16	853	29	0	3	4	17	852	29	0	3	4
	korku	11	8	1042	3	26	8	11	7	1044	3	26	7
	neşe	9	7	13	1231	13	35	9	7	13	1231	13	35
	üzüntü	39	9	38	14	1142	13	39	9	38	14	1143	12
	şaşıрма	14	7	15	39	14	662	14	7	15	39	14	662

Çizelge 4.23 içerisinde TREMO veri kümesi üzerinde NOEV yöntemi ve ArgMax yöntemiyle elde edilen sonuçlar doğruluk matrisi üzerinde gösterilmektedir. Önerilen NOEV yönteminin belirsizliğin yüksek olduğu “korku ve üzüntü” sınıflarında başarıyı arttırdığı diğer sınıflarda ise ArgMax yöntemiyle aynı sonuçları ürettiği görülmektedir.

Çizelge 4.24 TURTED Ön Eğitimli His Modeli ArgMax ve POEV doğruluk matrisi

His		ArgMax						POEV					
		öfke	tiksinme	korku	neşe	üzüntü	şaşıрма	öfke	tiksinme	korku	neşe	üzüntü	şaşıрма
Gerçek Etiket	öfke	8895	480	484	298	637	246	8867	480	486	305	654	248
	tiksinme	504	6470	336	299	388	168	493	6477	334	301	392	168
	korku	408	299	6413	311	437	307	402	298	6415	310	438	312
	neşe	400	209	436	5407	229	270	382	210	436	5419	231	273
	üzüntü	729	437	468	199	6512	240	710	437	471	197	6530	240
	şaşıрма	261	122	331	234	195	4803	255	118	332	234	200	4807

Çizelge 4.24 içerisinde TURTED veri kümesi üzerinde POEV yöntemi ve ArgMax yöntemiyle elde edilen sonuçlar doğruluk matrisi üzerinde gösterilmektedir. Önerilen POEV yöntemiyle görece daha büyük bir sınıf olan “öfke” haricindeki tüm sınıflarda doğru tahmin sayısı arttırılmıştır. Elde edilen deney sonuçlarına göre, klasik derin öğrenme yöntemleri üzerinde başarı gösteren Optimum His Vektörleri yönteminin (Çizelge 3.11), ön eğitimli his modeli ile çalıştırıldığında da belirsizliğin yüksek olduğu sınıflarda başarılı sonuçlar ürettiği görülmektedir. Bu sayede tez kapsamında önerilen Optimum His Vektörleri ve ön eğitimli his modeli yöntemlerinin bir birini tamamlayarak daha başarılı his analizi yaptığı sonucuna ulaşılmıştır.

5.SONUÇLAR VE GELECEK ÇALIŞMALAR

5.1. Sonuçlar

İnternet ve sayısal iletişimde yaşanan baş döndürücü teknolojik gelişmeler ve bunların etkisiyle oluşan toplumsal değişimler sonucunda ortaya çıkan büyük verinin his analizi yöntemleriyle damıtılarak kıymetlendirilmesi bir gereklilik haline gelmiştir. Bu tez kapsamında, bu gerekliliği yerine getirmek üzere, Türkçe yüksek performanslı his analizi gerçekleştirilmesi amaçlanmıştır. His analizinde en önemli bileşen öğrenme verisidir. Bu tezde öğrenme verilerinde karşılaşılan belirsizliği arttıran sınıflar arası dengesiz dağılım ve veri azlığı problemleri ele alınmıştır. Türkçe yüksek performanslı his analizi gerçekleştirmek amacıyla birbirini tamamlayan iki çalışma gerçekleştirilmiştir. Bu amaçla ele alınan ilk problem belirsizliğin yüksek olduğu dengesiz örnek uzayında daha etkili ve yüksek doğrulukta Türkçe his analizi yapmaktır. Çözüm olarak, optimizasyon yöntemleri yardımıyla derin öğrenme yaklaşımlarında belirsizliği azaltan bir şema önerilmektedir. Bu sayede görece daha az örneğe sahip olan küçük his sınıflarında daha başarılı his sınıflama yapılması amaçlanmaktadır. Bu işlem esnasında büyük sınıflarda elde edilen başarının korunması sağlanmaktadır.

Alan yazında yapılan inceleme sonucunda derin öğrenme yaklaşımlarının performansları değerlendirilirken sıklıkla genel doğruluk (*Accuracy*) metriğinin kullanıldığı tespit edilmiştir. Sadece genel doğruluk metriği kullanılarak sınıflama başarısı değerlendirildiğinde, veri içerisinde yer alan küçük sınıfların başarısı göz ardı edilmektedir. Bu tez çalışması kapsamında hacmine bakılmaksızın tüm his sınıflarında yüksek performans elde etmek amacıyla sınıf içi performanslar dikkate alınmaktadır. Önerilen optimum his vektörleri sayesinde klasik derin öğrenme tahmin mekanizmasına belirsizliği gideren bir yapı eklenmiştir. Bu sayede sadece en yüksek sınıfsal üyelik derecesini dikkate alan doğrusal ArgMax yönteminin yaptığı sınıflama hatası azaltılmaktadır. Önerilen yaklaşımla, özellikle bazı veri kümelerinde yer alan çok küçük his sınıflarında doğru tahmin oranı %46 arttırılmaktadır. Bu artış sınıflama yönteminin sınıf içi performanslarına olumlu katkı sağlamakta ve belirsizliği azaltmaktadır. Önerilen yaklaşım dile özgü öznitelikler kullanmadığından dilden bağımsız olarak çalışmaktadır. Bu sayede, tanımlanan belirsizliğin yüksek olduğu dengesiz örnek uzayında daha etkili ve yüksek doğrulukta his analizi yapma problemine bir çözüm üretilmektedir.

Hem geleneksel makine öğrenmesinin hem de derin öğrenme yöntemlerinin dezavantajlarını ortadan kaldırmak amacıyla ön eğitilmiş dil modelleri geliştirilmiştir. Bu yaklaşımda amaç bağlamdan bağımsız olarak büyük veri üzerinden genel amaçlı bir dil modeli elde etmek ve modelde yer alan ön bilgiyi alt probleme transfer ederek başarıyı arttırmaktır. Bu tez kapsamında çözüm önerilen ikinci problem ise az miktarda veri üzerinde ön eğitilmiş dil modeli kullanarak Türkçe his analizi alt yapısı oluşturmaktır. Bu amaçla çok dilli BERT ve Türkçe için hazırlanmış BERTurk ön eğitilmiş dil modelleri kullanılarak Türkçe his analizi gerçekleştirilmiştir. Önerilen yaklaşımla, geleneksel makine öğrenmesi sonuçlarında %8-%10, derin öğrenme sonuçlarında %2-%9 oranında artış sağlanmıştır. Ayrıca Türkçe his analizi için elde edilen TREMO için %92, TURTED için %79 sınıflama başarısı alan yazında rastlanan skorlara göre en yüksek sonuçlardır.

Bu tez çalışması kapsamında his analizinde performansı olumsuz etkileyen ve belirsizliği arttıran dengesiz veri dağılımı ve veri azlığı problemlerine birer çözüm önerilmektedir. Bu sayede hem belirsizliğin yüksek olduğu küçük his sınıflarının başarısı artırılmış hem de az veri bulunan Türkçe his analizinde en yüksek performans elde edilmiştir. Önerilen bir birini tamamlayan yöntemler birlikte kullanılarak yüksek performanslı his analizi gerçekleştirilmiştir.

Gerçekleştirilen yüksek performanslı his analizi yöntemleri Türkçe ve İngilizce kaynaklarda daha isabetli kamuoyu yoklaması, müşteri ilişkileri yönetimi, marka yönetimi, siber zorbalığın tespiti, seçimlerdeki yönelimin tespiti, partizan yorumları belirleme gibi üst seviye yapay zekâ görevlerin daha başarılı olarak yapılmasına olanak sağlayacaktır. Ayrıca bu alandaki araştırmacılar için bir kaynak olacak Türkçe metin madenciliği görevlerinde kullanılabilir.

5.2. Gelecek Çalışmalar

Bu tez çalışması kapsamında ortaya atılan optimum his vektörleri, derin öğrenme modelinden elde edilen öğrenme kümesi çıktılarından faydalanarak optimizasyon yöntemleriyle elde edilmektedir. Mevcut durumda önce derin öğrenme yöntemi

işletilmekte ve ardından optimizasyon algoritması bağımsız olarak çalışarak optimum his vektörlerini üretmektedir. İleriki çalışmalarda kullanılan derin öğrenme kütüphanelerinin elverdiği ölçüde bu iki yöntem entegre edilerek birlikte çalıştırılması planlanmaktadır. Bu sayede oluşturulacak çerçeve alternatif veri kümelerine uygulanabileceği gibi başka problemlere de uyarlanabilecektir.

Öte yandan önerilen yöntemde oluşturulan optimum his vektörleri, ilgili öğrenme kümesi üzerinden elde edildiği için bağlama bağlı kalmakta ve alternatif veri kümeleri üzerinde kullanılamamaktadır. İleride, elde edilen tüm veri kümeleri birleştirilerek etiketli bir his derlemi oluşturulup bağlamdan bağımsız optimum his vektörleri üretmek planlanan çalışmalar arasındadır.

Öte yandan mevcut alt yapı kullanılarak temel hislerin yanı sıra karmaşık hislerin yer aldığı veri kümeleri üzerinde his analizi yapılabilir. Alternatif olarak önerilen optimum his vektörleri yaklaşımı derin öğrenme ağının gizli katmanlarından gelen bilgilerle kullanılarak yeni bir sınıflayıcı tasarlanabilir.

Gelecekte, geliştirilen ön eğitilmiş dil modeli adaptasyon mekanizmasıyla önce duygu analizi gibi yakın bağlamlarda ince-ayar yapılarak sonra his analizi bağlamına ince-ayar yapılarak ek bir bilgi transfer edilmesi planlanmaktadır. Ayrıca temel hisler üzerinde ince-ayar yapılan model ile gözetimsiz olarak karmaşık his verileri için çok etiketli his analizi yapılması planlar arasındadır. Ek olarak Türkçe karmaşık hisler veri kümesi oluşturmak için sosyal medya platformlarından veri toplanarak paylaşılması planlanmaktadır.

5. KAYNAKLAR

- [1] Y. Goldberg, Neural network methods for natural language processing, *Synth. Lect. Hum. Lang. Technol.* 10 (2017) 1–309.
- [2] D. Jurafsky, J.H. Martin, *Speech and language processing: an introduction to natural language processing, computational linguistics, and speech recognition*, (2000).
- [3] S. Aman, S. Szpakowicz, Identifying expressions of emotion in text, in: *Int. Conf. Text, Speech Dialogue*, 2007: pp. 196–205.
- [4] T. Mikolov, G. Corrado, K. Chen, J. Dean, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, *Proc. Int. Conf. Learn. Represent. (ICLR 2013)*. (2013) 1–12.
- [5] J. Pennington, R. Socher, C. Manning, Glove: Global Vectors for Word Representation, in: *Proc. 2014 Conf. Empir. Methods Nat. Lang. Process.*, Association for Computational Linguistics, Stroudsburg, PA, USA, 2014: pp. 1532–1543. doi:10.3115/v1/D14-1162.
- [6] P. Bojanowski, E. Grave, A. Joulin, T. Mikolov, Enriching word vectors with subword information, *Trans. Assoc. Comput. Linguist.* 5 (2017) 135–146.
- [7] Q. V. Le, T. Mikolov, Distributed Representations of Sentences and Documents, 32 (2014). doi:10.1145/2740908.2742760.
- [8] Z. Lin, M. Feng, C.N. dos Santos, M. Yu, B. Xiang, B. Zhou, Y. Bengio, A structured self-attentive sentence embedding, *ArXiv Prepr. ArXiv1703.03130*. (2017).
- [9] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, (2018). doi:arXiv:1811.03600v2.
- [10] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R.R. Salakhutdinov, Q. V Le, Xlnet: Generalized autoregressive pretraining for language understanding, in: *Adv. Neural Inf. Process. Syst.*, 2019: pp. 5753–5763.
- [11] H. Al-omari, M.A. Abdullah, S. Shaikh, EmoDet2 : Emotion Detection in English

- Textual Dialogue using BERT and BiLSTM Models, (2020) 226–232.
- [12] S. Schweter, BERTurk - BERT models for Turkish, (2020). doi:10.5281/zenodo.3770924.
- [13] P. Ekman, Universals and cultural differences in facial expressions of emotion, in: Nebraska Symp. Motiv., 1972: pp. 207–282.
- [14] P. Ekman, W. V. Friesen, Measuring facial movement, Environ. Psychol. Nonverbal Behav. 1 (1976) 56–75. doi:10.1007/BF01115465.
- [15] C. Darwin, The Expression of the Emotions in Man and Animals, 1873.
- [16] P. Ekman, An argument for basic emotions, Cogn. Emot. 6 (1992) 169–200.
- [17] R. Plutchik, The emotions, University Press of America, 1991.
- [18] J.A. Russell, G. Pratt, A description of the affective quality attributed to environments., J. Pers. Soc. Psychol. 38 (1980) 311.
- [19] J. Hellrich, S. Buechel, U. Hahn, Modeling Word Emotion in Historical Language: Quantity Beats Supposed Stability in Seed Word Selection, (2018).
- [20] W. Amelia, N.U. Maulidevi, Dominant Emotion Recognition in Short Story Using Keyword Spotting Technique and Learning-based Method, in: 2016 Int. Conf. Adv. INFORMATICS - CONCEPTS, THEORY Appl., 2016.
- [21] M.J.C. Samonte, H.I.B. Punzalan, R.J.P.G. Santiago, P.J.L. Linchangco, Emotion detection in blog posts using keyword spotting and semantic analysis, in: Proc. 3rd Int. Conf. Commun. Inf. Process., 2017: pp. 6–13.
- [22] C. Ma, H. Prendinger, M. Ishizuka, Emotion estimation and reasoning based on affective textual interaction, in: Int. Conf. Affect. Comput. Intell. Interact., 2005: pp. 622–628.
- [23] H. Binali, C. Wu, V. Potdar, Computational approaches for emotion detection in text, in: 4th IEEE Int. Conf. Digit. Ecosyst. Technol., IEEE, 2010: pp. 172–177. doi:10.1109/DEST.2010.5610650.
- [24] I. Perikos, I. Hatzilygeroudis, Recognizing emotion presence in natural language sentences, in: Int. Conf. Eng. Appl. Neural Networks, 2013: pp. 30–39.
- [25] S.M. Mohammad, P.D. Turney, Crowdsourcing a word-emotion association lexicon, Comput. Intell. 29 (2013) 436–465. doi:10.1111/j.1467-

8640.2012.00460.x.

- [26] A. Bandhakavi, N. Wiratunga, S. Massie, D. Padmanabhan, Lexicon Generation for Emotion Detection from Text, *IEEE Intell. Syst.* 32 (2017) 102–108.
- [27] S.M. Mohammad, P.D. Turney, Emotions evoked by common words and phrases: Using Mechanical Turk to create an emotion lexicon, in: *Proc. NAACL HLT 2010 Work. Comput. Approaches to Anal. Gener. Emot. Text*, 2010: pp. 26–34.
- [28] P.J. Stone, D.C. Dunphy, M.S. Smith, *The general inquirer: A computer approach to content analysis.*, (1966).
- [29] C. Strapparava, A. Valitutti, others, WordNet Affect: an Affective Extension of WordNet., in: *LREC*, 2004: pp. 1083–1086.
- [30] S.M. Mohammad, Obtaining Reliable Human Ratings of Valence, Arousal, and Dominance for 20,000 English Words, in: *Proc. Annu. Conf. Assoc. Comput. Linguist.*, Melbourne, Australia, 2018.
- [31] S.M. Mohammad, S. Kiritchenko, USING HASHTAGS TO CAPTURE FINE EMOTION CATEGORIES FROM TWEETS, *Comput. Intell.* 31 (2015) 301–326. doi:10.1111/coin.12024.
- [32] M.A. Tocoglu, A. Alpkocak, TREMO: A dataset for emotion analysis in Turkish, *J. Inf. Sci.* 44 (2018) 848–860.
- [33] M.A. Tocoglu, A. Alpkocak, Lexicon-based emotion analysis in Turkish, *Turkish J. Electr. Eng. Comput. Sci.* 27 (2019) 1213–1227.
- [34] R. Kamal, M.A. Shah, C. Maple, M. Masood, A. Wahid, A. Mehmood, Emotion classification and crowd source sensing; a lexicon based approach, *IEEE Access.* 7 (2019) 27124–27134.
- [35] A. Bandhakavi, N. Wiratunga, D. Padmanabhan, S. Massie, Lexicon based feature extraction for emotion text classification, *Pattern Recognit. Lett.* 93 (2017) 133–142. doi:10.1016/j.patrec.2016.12.009.
- [36] A. Purpura, C. Masiero, G. Silvello, G.A. Susto, Feature Selection for Emotion Classification., in: *IIR*, 2019: pp. 47–48.
- [37] P. Katz, M. Singleton, R. Wicentowski, Swat-mp: the semeval-2007 systems for task 5 and task 14, in: *Proc. Fourth Int. Work. Semant. Eval.*, 2007: pp. 308–313.

- [38] J. Yang, L. Jiang, C. Wang, J. Xie, Multi-Label Emotion Classification for Tweets in Weibo: Method and Application, in: 2014 IEEE 26TH Int. Conf. TOOLS WITH Artif. Intell., 2014: pp. 424–428. doi:10.1109/ICTAI.2014.71.
- [39] F. Provost, R. Kohavi, Glossary of terms, *J. Mach. Learn.* 30 (1998) 271–274.
- [40] S. Samothrakis, M. Fasli, Emotional sentence annotation helps predict fiction genre, *PLoS One.* 10 (2015) e0141922.
- [41] E. Kim, S. Padó, R. Klinger, Investigating the relationship between literary genres and emotional plot development, in: *Proc. Jt. SIGHUM Work. Comput. Linguist. Cult. Heritage, Soc. Sci. Humanit. Lit.*, 2017: pp. 17–26.
- [42] S. Aman, S. Szpakowicz, Using roget's thesaurus for fine-grained emotion recognition, in: *Proc. Third Int. Jt. Conf. Nat. Lang. Process. Vol.*, 2008.
- [43] D. Ghazi, D. Inkpen, S. Szpakowicz, Hierarchical versus flat classification of emotions in text, in: *Proc. NAACL HLT 2010 Work. Comput. Approaches to Anal. Gener. Emot. Text*, 2010: pp. 140–146.
- [44] S. Mac Kim, A. Valitutti, R. a Calvo, Evaluation of Unsupervised Emotion Models to Textual Affect Recognition, *Proc. NAACL HLT 2010 Work. Comput. Approaches to Anal. Gener. Emot. Text.* (2010) 62–70.
- [45] S. Chaffar, D. Inkpen, Using a heterogeneous dataset for emotion analysis in text, in: *Can. Conf. Artif. Intell.*, 2011: pp. 62–67.
- [46] D. Ghazi, D. Inkpen, S. Szpakowicz, Prior and contextual emotion of words in sentential context, *Comput. Speech Lang.* 28 (2014) 76–92. doi:10.1016/j.csl.2013.04.009.
- [47] F. Akba, A. Uçan, E. Sezer, H. Sever, Assessment of feature selection metrics for sentiment analyses: Turkish movie reviews, *8th Eur. Conf. Data Min.* 2014. 191 (2014) 180–184.
- [48] B. Yu, An evaluation of text classification methods for literary study, *Lit. Linguist. Comput.* 23 (2008) 327–343.
- [49] C. Cortes, V. Vapnik, Support-vector networks, *Mach. Learn.* 20 (1995) 273–297. doi:10.1007/BF00994018.
- [50] G.I. Webb, Naïve Bayes, in: C. Sammut, G.I. Webb (Eds.), *Encycl. Mach. Learn.*,

- Springer US, Boston, MA, 2010: pp. 713–714. doi:10.1007/978-0-387-30164-8_576.
- [51] S.R. Safavian, D. Landgrebe, A survey of decision tree classifier methodology, *IEEE Trans. Syst. Man. Cybern.* 21 (1991) 660–674.
- [52] C. Strapparava, R. Mihalcea, Learning to identify emotions in text, in: *Proc. 2008 ACM Symp. Appl. Comput.*, 2008: pp. 1556–1560.
- [53] C.O. Alm, D. Roth, R. Sproat, Emotions from text: machine learning for text-based emotion prediction, *Proc. Hum. Lang. Technol. Conf. Conf. Empir. Methods Nat. Lang. Process.* (2005) 579–586. doi:10.3115/1220575.1220648.
- [54] A. Carlson, C. Cumby, J. Rosen, D. Roth, The SNoW learning architecture, n.d.
- [55] R.C. Balabantaray, M. Mohammad, N. Sharma, Multi-class twitter emotion classification: A new approach, *Int. J. Appl. Inf. Syst.* 4 (2012) 48–53.
- [56] K. Roberts, M.A. Roach, J. Johnson, J. Guthrie, S.M. Harabagiu, EmpaTweet: Annotating and Detecting Emotions on Twitter., in: *Lrec, 2012*: pp. 3806–3813.
- [57] T. Danisman, A. Alpkocak, Feeler: Emotion Classification of Text Using Vector Space Model, *Aisb.* (2008) 53–59.
- [58] K.R. Scherer, H.G. Wallbott, Evidence for universality and cultural variation of differential emotion response patterning., *J. Pers. Soc. Psychol.* 66 (1994) 310.
- [59] A. Khan, B. Baharudin, L.H. Lee, K. Khan, A review of machine learning algorithms for text-documents classification, *J. Adv. Inf. Technol.* 1 (2010) 4–20.
- [60] K. Sailunaz, M. Dhaliwal, J. Rokne, R. Alhajj, Emotion detection from text and speech: a survey, *Soc. Netw. Anal. Min.* 8 (2018). doi:10.1007/s13278-018-0505-2.
- [61] K. Suzuki, *Artificial neural networks: architectures and applications*, BoD--Books on Demand, 2013.
- [62] W.S. McCulloch, W. Pitts, A logical calculus of the ideas immanent in nervous activity, *Bull. Math. Biophys.* 5 (1943) 115–133.
- [63] F. Rosenblatt, The perceptron: a probabilistic model for information storage and organization in the brain., *Psychol. Rev.* 65 (1958) 386.
- [64] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning internal representations by

- error propagation, 1985.
- [65] G.E. Hinton, S. Osindero, Y.-W. Teh, A fast learning algorithm for deep belief nets, *Neural Comput.* 18 (2006) 1527–1554.
 - [66] Y. Bengio, P. Lamblin, D. Popovici, H. Larochelle, Greedy layer-wise training of deep networks, in: *Adv. Neural Inf. Process. Syst.*, 2007: pp. 153–160.
 - [67] Y. Bengio, A. Courville, P. Vincent, Representation learning: A review and new perspectives, *IEEE Trans. Pattern Anal. Mach. Intell.* 35 (2013) 1798–1828.
 - [68] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, *Proc. IEEE.* 86 (1998) 2278–2324.
 - [69] L. Mosser, O. Dubrule, M.J. Blunt, Stochastic Reconstruction of an Oolitic Limestone by Generative Adversarial Networks, *Transp. Porous Media.* 125 (2018) 81–103. doi:10.1007/s11242-018-1039-9.
 - [70] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: *Adv. Neural Inf. Process. Syst.*, 2012: pp. 1097–1105.
 - [71] A. Graves, A. Mohamed, G. Hinton, Speech recognition with deep recurrent neural networks, in: *2013 IEEE Int. Conf. Acoust. Speech Signal Process.*, 2013: pp. 6645–6649.
 - [72] Y. Kim, *Convolutional Neural Networks for Sentence Classification*, (2014). doi:10.3115/v1/D14-1181.
 - [73] Y. Zhang, B. Wallace, A sensitivity analysis of (and practitioners’ guide to) convolutional neural networks for sentence classification, *ArXiv Prepr. ArXiv1510.03820.* (2015).
 - [74] C.M. Bishop, *Pattern recognition and machine learning*, Springer-Verlag New York, 2006.
 - [75] S.M. Zahiri, J.D. Choi, Emotion detection on tv show transcripts with sequence-based convolutional neural networks, in: *Work. Thirty-Second AAAI Conf. Artif. Intell.*, 2018.
 - [76] S. Khosla, Emotionx-ar: CNN-DCNN autoencoder based emotion classifier, in: *Proc. Sixth Int. Work. Nat. Lang. Process. Soc. Media*, 2018: pp. 37–44.

- [77] A. Zeyer, P. Doetsch, P. Voigtlaender, R. Schlüter, H. Ney, A comprehensive study of deep bidirectional LSTM RNNs for acoustic modeling in speech recognition, in: 2017 IEEE Int. Conf. Acoust. Speech Signal Process., 2017: pp. 2462–2466.
- [78] S. Kombrink, T. Mikolov, M. Karafiát, L. Burget, Recurrent neural network based language modeling in meeting recognition, in: Twelfth Annu. Conf. Int. Speech Commun. Assoc., 2011.
- [79] I. Sutskever, O. Vinyals, Q. V Le, Sequence to sequence learning with neural networks, in: Adv. Neural Inf. Process. Syst., 2014: pp. 3104–3112.
- [80] K. Cho, B. Van Merriënboer, D. Bahdanau, Y. Bengio, On the properties of neural machine translation: Encoder-decoder approaches, ArXiv Prepr. ArXiv1409.1259. (2014).
- [81] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, *Nature*. 521 (2015) 436–444.
- [82] Y. Bengio, P. Simard, P. Frasconi, Learning long-term dependencies with gradient descent is difficult, *IEEE Trans. Neural Networks*. 5 (1994) 157–166.
- [83] S. Hochreiter, J.J. Schmidhuber, Long short-term memory, *Neural Comput.* 9 (1997) 1–32.
- [84] M. Schuster, K.K. Paliwal, Bidirectional recurrent neural networks, *IEEE Trans. Signal Process.* 45 (1997) 2673–2681.
- [85] J. Schmidhuber, F. Gers, D. Eck, Learning nonregular languages: A comparison of simple recurrent networks and LSTM, *Neural Comput.* 14 (2002) 2039–2041.
- [86] A. Graves, J. Schmidhuber, Offline handwriting recognition with multidimensional recurrent neural networks, in: Adv. Neural Inf. Process. Syst., 2009: pp. 545–552.
- [87] A. Graves, J. Schmidhuber, Framewise phoneme classification with bidirectional LSTM and other neural network architectures, *Neural Networks*. 18 (2005) 602–610.
- [88] S. Fernández, A. Graves, J. Schmidhuber, An application of recurrent neural networks to discriminative keyword spotting, in: Int. Conf. Artif. Neural Networks, 2007: pp. 220–229.
- [89] M. Köper, E. Kim, R. Klinger, IMS at EmoInt-2017: Emotion Intensity Prediction

- with Affective Norms, Automatically Extended Resources and Deep Learning, (2017) 50–57.
- [90] B. Naderalvojud, A. Ucan, E. Akcapinar Sezer, HUMIR at IEST-2018: Lexicon-sensitive and left-right context-sensitive bi-lstm for implicit emotion recognition, in: Proc. 9th Work. Comput. Approaches to Subj. Sentim. Soc. Media Anal., Association for Computational Linguistics, 2018: pp. 182–188.
- [91] D. Haryadi, G.P. Kusuma, Emotion Detection in Text Using Nested Long Short-Term Memory, *IJACSA Int. J. Adv. Comput. Sci. Appl.* 10 (2019).
- [92] H. Ren, J. Wan, Y. Ren, Emotion detection in cross-lingual text based on bidirectional LSTM, in: *Int. Conf. Secur. with Intell. Comput. Big-Data Serv.*, 2018: pp. 838–845.
- [93] U. Gupta, A. Chatterjee, R. Srikanth, P. Agrawal, A Sentiment-and-Semantics-Based Approach for Emotion Detection in Textual Conversations, (2017).
- [94] J. Kim, H. Kim, others, Classification performance using gated recurrent unit recurrent neural network on energy disaggregation, in: *2016 Int. Conf. Mach. Learn. Cybern.*, 2016: pp. 105–110.
- [95] Y. Su, C.-C.J. Kuo, On extended long short-term memory and dependent bidirectional recurrent neural network, *Neurocomputing*. 356 (2019) 151–161. doi:10.1016/j.neucom.2019.04.044.
- [96] G. Weiss, Y. Goldberg, E. Yahav, On the Practical Computational Power of Finite Precision RNNs for Language Recognition, (2018).
- [97] D. Britz, A. Goldie, M.-T. Luong, Q. Le, Massive Exploration of Neural Machine Translation Architectures, (2017).
- [98] M. Ravanelli, P. Brakel, M. Omologo, Y. Bengio, Light Gated Recurrent Units for Speech Recognition, *IEEE Trans. Emerg. Top. Comput. Intell.* 2 (2018) 92–102. doi:10.1109/tetci.2017.2762739.
- [99] J. Chung, C. Gulcehre, K. Cho, Y. Bengio, Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling, (2014).
- [100] N. Gruber, A. Jockisch, Are GRU Cells More Specific and LSTM Cells More Sensitive in Motive Classification of Text?, *Front. Artif. Intell.* 3 (2020) 40.

doi:10.3389/frai.2020.00040.

- [101] D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, ArXiv Prepr. ArXiv1409.0473. (2014).
- [102] P. Liu, X. Qiu, X. Huang, Recurrent neural network for text classification with multi-task learning, ArXiv Prepr. ArXiv1605.05101. (2016).
- [103] M. Abdul-Mageed, L. Ungar, EmoNet: Fine-Grained Emotion Detection with Gated Recurrent Neural Networks, Proc. 55th Annu. Meet. Assoc. for Comput. Linguist. (Volume 1 Long Pap. (2017) 718–728. doi:10.18653/v1/P17-1067.
- [104] W. Jiao, H. Yang, I. King, M.R. Lyu, HiGRU: Hierarchical Gated Recurrent Units for Utterance-level Emotion Recognition, (2019).
- [105] P. Rathnayaka, S. Abeysinghe, C. Samarajeewa, I. Manchanayake, M.J. Walpola, R. Nawaratne, T. Bandaragoda, D. Alahakoon, Gated Recurrent Neural Network Approach for Multilabel Emotion Detection in Microblogs, (2019).
- [106] T. Tom´, T. Mikolov, I. Sutskever, K. Chen, G. Corrado, J. Dean, Q. Le, T. Strohmann, G. Research, Learning Representations of Text using Neural Networks NIPS Deep Learning Workshop 2013, (2013) 1–31.
- [107] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask R-CNN, (2017).
- [108] H. Zhao, J. Shi, X. Qi, X. Wang, J. Jia, Pyramid Scene Parsing Network, (2016).
- [109] G. Papandreou, T. Zhu, N. Kanazawa, A. Toshev, J. Tompson, C. Bregler, K. Murphy, Towards Accurate Multi-person Pose Estimation in the Wild, (2017).
- [110] J. Carreira, A. Zisserman, Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset, (2017).
- [111] G. Zweig, C.J.C. Burges, The Microsoft Research Sentence Completion Challenge, 2011.
- [112] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, C. Dyer, Neural architectures for named entity recognition, ArXiv Prepr. ArXiv1603.01360. (2016).
- [113] T. Mikolov, Q. V Le, I. Sutskever, Exploiting similarities among languages for machine translation, ArXiv Prepr. ArXiv1309.4168. (2013).
- [114] H. Wang, N. Wang, D.-Y. Yeung, Collaborative deep learning for recommender

- systems, in: Proc. 21th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min., 2015: pp. 1235–1244.
- [115] A. Chatterjee, U. Gupta, M.K. Chinnakotla, R. Srikanth, M. Galley, P. Agrawal, Understanding emotions in text using deep learning and big data, *Comput. Human Behav.* 93 (2018) 309–317. doi:10.1016/J.CHB.2018.12.029.
- [116] A. Agrawal, A. An, M. Papagelis, Learning Emotion-enriched Word Representations, *Proc. 27th Int. Conf. Comput. Linguist.* (2018) 950–961.
- [117] S. Ghosh, M. Chollet, E. Laksana, L.-P. Morency, S. Scherer, Affect-lm: A neural language model for customizable affective text generation, *ArXiv Prepr. ArXiv1704.06851.* (2017).
- [118] A. Seyeditabari, W. Zadrozny, Can word embeddings help find latent emotions in text? preliminary results, in: *Thirtieth Int. Flairs Conf.*, 2017.
- [119] J. Savigny, A. Purwarianti, Emotion classification on youtube comments using word embedding, in: *2017 Int. Conf. Adv. Informatics, Concepts, Theory, Appl.*, 2017: pp. 1–5.
- [120] B. Naderalvojud, A. Ucan, E.A. Sezer, HUMIR at IEST-2018 : Lexicon-Sensitive and Left-Right Context-Sensitive BiLSTM for Implicit Emotion Recognition, (2018) 182–188.
- [121] D. Tang, B. Qin, T. Liu, Document modeling with gated recurrent neural network for sentiment classification, in: *Proc. 2015 Conf. Empir. Methods Nat. Lang. Process.*, 2015: pp. 1422–1432.
- [122] J.P.C. Chiu, E. Nichols, Named entity recognition with bidirectional LSTM-CNNs, *Trans. Assoc. Comput. Linguist.* 4 (2016) 357–370.
- [123] A. Conneau, G. Lample, M. Ranzato, L. Denoyer, H. Jégou, Word translation without parallel data, *ArXiv Prepr. ArXiv1710.04087.* (2017).
- [124] C. Xiong, V. Zhong, R. Socher, Dynamic coattention networks for question answering, *ArXiv Prepr. ArXiv1611.01604.* (2016).
- [125] A. Chatterjee, K.N. Narahari, M. Joshi, P. Agrawal, SemEval-2019 Task 3: EmoContext Contextual Emotion Detection in Text, in: *Proc. 13th Int. Work. Semant. Eval.*, 2019: pp. 39–48.

- [126] L. Rheault, K. Beelen, C. Cochrane, G. Hirst, Measuring emotion in parliamentary debates with automated textual analysis, *PLoS One*. 11 (2016) e0168843.
- [127] A. Agrawal, A. An, M. Papagelis, Learning emotion-enriched word representations, in: *Proc. 27th Int. Conf. Comput. Linguist.*, 2018: pp. 950–961.
- [128] F. Krebs, B. Lubascher, T. Moers, P. Schaap, G. Spanakis, Social emotion mining techniques for Facebook posts reaction prediction, *ArXiv Prepr. ArXiv1712.03249*. (2017).
- [129] X. Li, Y. Rao, H. Xie, R.Y.K. Lau, J. Yin, F.L. Wang, Bootstrapping social emotion classification with semantically rich hybrid neural networks, *IEEE Trans. Affect. Comput.* 8 (2017) 428–442.
- [130] S. Ge, T. Qi, C. Wu, Y. Huang, THU_NGN at SemEval-2019 Task 3: Dialog Emotion Classification using Attentional LSTM-CNN, in: *Proc. 13th Int. Work. Semant. Eval.*, 2019: pp. 340–344.
- [131] G. Lample, A. Conneau, L. Denoyer, M. Ranzato, Unsupervised machine translation using monolingual corpora only, *ArXiv Prepr. ArXiv1711.00043*. (2017).
- [132] N. Reimers, I. Gurevych, Reporting score distributions makes a difference: Performance study of lstm-networks for sequence tagging, *ArXiv Prepr. ArXiv1707.09861*. (2017).
- [133] A. Fan, M. Lewis, Y. Dauphin, Hierarchical neural story generation, *ArXiv Prepr. ArXiv1805.04833*. (2018).
- [134] S. Reddy, D. Chen, C.D. Manning, Coqa: A conversational question answering challenge, *Trans. Assoc. Comput. Linguist.* 7 (2019) 249–266.
- [135] S. Buechel, U. Hahn, Emotion representation mapping for automatic lexicon construction (mostly) performs on human level, *ArXiv Prepr. ArXiv1806.08890*. (2018).
- [136] E. Saravia, H.-C.T. Liu, Y.-H. Huang, J. Wu, Y.-S. Chen, Carer: Contextualized affect representations for emotion recognition, in: *Proc. 2018 Conf. Empir. Methods Nat. Lang. Process.*, 2018: pp. 3687–3697.
- [137] S. Buechel, U. Hahn, Word emotion induction for multiple languages as a deep

- multi-task learning problem, in: Proc. 2018 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. Vol. 1 (Long Pap., 2018: pp. 1907–1918.
- [138] R.E.N. Fuji, K. Matsumoto, Emotion analysis on social big data, *ZTE Commun.* 15 (2020) 30–37.
- [139] M. Jabreel, A. Moreno, EiTAKA at SemEval-2018 Task 1: An ensemble of n-channels ConvNet and XGboost regressors for emotion analysis of tweets, *ArXiv Prepr. ArXiv1802.09233.* (2018).
- [140] Z. Ahmad, R. Jindal, A. Ekbal, P. Bhattacharyya, Borrow from rich cousin: transfer learning for emotion detection using cross lingual embedding, *Expert Syst. Appl.* 139 (2020) 112851. doi:10.1016/j.eswa.2019.112851.
- [141] M.E. Peters, M. Neumann, L. Zettlemoyer, W. Yih, Dissecting contextual word embeddings: Architecture and representation, *ArXiv Prepr. ArXiv1808.08949.* (2018).
- [142] X. Qiu, T. Sun, Y. Xu, Y. Shao, N. Dai, X. Huang, Pre-trained Models for Natural Language Processing: A Survey, (2020).
- [143] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, I. Polosukhin, Attention Is All You Need, (2017).
- [144] V. Sanh, L. Debut, J. Chaumond, T. Wolf, DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter, *ArXiv Prepr. ArXiv1910.01108.* (2019).
- [145] M. Zampieri, S. Malmasi, P. Nakov, S. Rosenthal, N. Farra, R. Kumar, Semeval-2019 task 6: Identifying and categorizing offensive language in social media (offenseval), *ArXiv Prepr. ArXiv1903.08983.* (2019).
- [146] R. Zellers, A. Holtzman, H. Rashkin, Y. Bisk, A. Farhadi, F. Roesner, Y. Choi, Defending against neural fake news, in: *Adv. Neural Inf. Process. Syst.*, 2019: pp. 9054–9065.
- [147] Y. Zhang, X. Chen, Explainable recommendation: A survey and new perspectives, *ArXiv Prepr. ArXiv1804.11192.* (2018).
- [148] Y. Liu, M. Lapata, Text summarization with pretrained encoders, *ArXiv Prepr. ArXiv1908.08345.* (2019).

- [149] A. Chatterjee, N. Narahari, M. Joshi, P. Agrawal, SemEval-2019 Task 3: EmoContext Contextual Emotion Detection in Text, n.d.
- [150] J. Xiao, Figure eight at SemEval-2019 task 3: ensemble of transfer learning methods for contextual emotion detection, in: Proc. 13th Int. Work. Semant. Eval., 2019: pp. 220–224.
- [151] P. Zhong, C. Miao, ntuer at semeval-2019 task 3: Emotion classification with word and sentence representations in rcnn, ArXiv Prepr. ArXiv1902.07867. (2019).
- [152] L. Ma, L. Zhang, W. Ye, W. Hu, PKUSE at SemEval-2019 task 3: emotion detection with emotion-oriented neural attention network, in: Proc. 13th Int. Work. Semant. Eval., 2019: pp. 287–291.
- [153] P. Zhong, D. Wang, C. Miao, Knowledge-enriched transformer for emotion detection in textual conversations, ArXiv Prepr. ArXiv1909.10681. (2019).
- [154] Y.-H. Huang, S.-R. Lee, M.-Y. Ma, Y.-H. Chen, Y.-W. Yu, Y.-S. Chen, EmotionX-IDEA: Emotion BERT--an Affectional Model for Conversation, ArXiv Prepr. ArXiv1908.06264. (2019).
- [155] A. Basile, M. Franco-Salvador, N. Pawar, S. Štajner, M.C. Rios, Y. Benajiba, SymantoResearch at SemEval-2019 task 3: combined neural models for emotion classification in human-chatbot conversations, in: Proc. 13th Int. Work. Semant. Eval., 2019: pp. 330–334.
- [156] L. Parisi, S. Francia, S. Olivastri, M.S. Tavella, Exploiting synchronized lyrics and vocal features for music emotion detection, ArXiv Prepr. ArXiv1901.04831. (2019).
- [157] N. Colnerič, Emotion recognition on twitter using neural networks, Univerza v Ljubljani, Fakulteta za računalništvo in informatiko, 2019.
- [158] H. Tang, D. Ji, Q. Zhou, Joint Multi-level Attentional Model for Emotion Detection and Emotion-cause Pair Extraction, Neurocomputing. (2020).
- [159] H. Fei, D. Ji, Y. Zhang, Y. Ren, Topic-Enhanced Capsule Network for Multi-Label Emotion Classification, IEEE/ACM Trans. Audio, Speech, Lang. Process. 28 (2020) 1839–1848.
- [160] M.E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, L. Zettlemoyer,

- Deep contextualized word representations, (2018).
- [161] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, Improving language understanding by generative pre-training, (2018).
- [162] L. Kong, C. de M. d’Autume, W. Ling, L. Yu, Z. Dai, D. Yogatama, A Mutual Information Maximization Perspective of Language Representation Learning, ArXiv Prepr. ArXiv1910.08350. (2019).
- [163] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, Roberta: A robustly optimized bert pretraining approach, ArXiv Prepr. ArXiv1907.11692. (2019).
- [164] K. Clark, M.-T. Luong, Q. V Le, C.D. Manning, Electra: Pre-training text encoders as discriminators rather than generators, ArXiv Prepr. ArXiv2003.10555. (2020).
- [165] L. Dong, N. Yang, W. Wang, F. Wei, X. Liu, Y. Wang, J. Gao, M. Zhou, H.-W. Hon, Unified language model pre-training for natural language understanding and generation, in: Adv. Neural Inf. Process. Syst., 2019: pp. 13063–13075.
- [166] K. Song, X. Tan, T. Qin, J. Lu, T.-Y. Liu, Mass: Masked sequence to sequence pre-training for language generation, ArXiv Prepr. ArXiv1905.02450. (2019).
- [167] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, L. Zettlemoyer, Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension, ArXiv Prepr. ArXiv1910.13461. (2019).
- [168] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P.J. Liu, Exploring the limits of transfer learning with a unified text-to-text transformer, ArXiv Prepr. ArXiv1910.10683. (2019).
- [169] Z. Zhang, X. Han, Z. Liu, X. Jiang, M. Sun, Q. Liu, ERNIE: Enhanced language representation with informative entities, ArXiv Prepr. ArXiv1905.07129. (2019).
- [170] X. Jiao, Y. Yin, L. Shang, X. Jiang, X. Chen, L. Li, F. Wang, Q. Liu, Tinybert: Distilling bert for natural language understanding, ArXiv Prepr. ArXiv1909.10351. (2019).
- [171] S. Sun, Y. Cheng, Z. Gan, J. Liu, Patient knowledge distillation for bert model compression, ArXiv Prepr. ArXiv1908.09355. (2019).

- [172] Z. Sun, H. Yu, X. Song, R. Liu, Y. Yang, D. Zhou, Mobilebert: a compact task-agnostic bert for resource-limited devices, *ArXiv Prepr. ArXiv2004.02984*. (2020).
- [173] W. Wang, F. Wei, L. Dong, H. Bao, N. Yang, M. Zhou, Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers, *ArXiv Prepr. ArXiv2002.10957*. (2020).
- [174] S. Zhao, R. Gupta, Y. Song, D. Zhou, Extreme language model compression with optimal subwords and shared projections, *ArXiv Prepr. ArXiv1909.11687*. (2019).
- [175] W. Antoun, F. Baly, H. Hajj, AraBERT: Transformer-based Model for Arabic Language Understanding, (2020).
- [176] Y. Cui, W. Che, T. Liu, B. Qin, Z. Yang, S. Wang, G. Hu, Pre-Training with Whole Word Masking for Chinese BERT, (2019).
- [177] A. Virtanen, J. Kanerva, R. Ilo, J. Luoma, J. Luotolahti, T. Salakoski, F. Ginter, S. Pyysalo, Multilingual is not enough: BERT for Finnish, (2019).
- [178] L. Martin, B. Muller, P.J.O. Suárez, Y. Dupont, L. Romary, É.V. de la Clergerie, D. Seddah, B. Sagot, CamemBERT: a Tasty French Language Model, (2019).
- [179] Y. Kikuta, BERT Pretrained model Trained On Japanese Wikipedia Articles, *GitHub Repos*. (2019).
- [180] Y. Kuratov, M. Arkhipov, Adaptation of Deep Bidirectional Multilingual Transformers for Russian Language, (2019).
- [181] X.-S. Yang, Firefly algorithm, stochastic test functions and design optimisation, *Int. J. Bio-Inspired Comput.* 2 (2010) 78–84.
- [182] J. Kennedy, R. Eberhart, Particle swarm optimization, in: *Proc. ICNN'95-International Conf. Neural Networks, 1995*: pp. 1942–1948.
- [183] S.M. Mirjalili, S.M. Mirjalili, A. Lewis, Grey Wolf Optimizer, *Adv. Eng. Softw.* 69 (2014) 46–61. doi:10.1016/j.advengsoft.2013.12.007.
- [184] E. Emary, H.M. Zawbaa, A.E. Hassanien, Binary grey wolf optimization approaches for feature selection, *Neurocomputing.* 172 (2016) 371–381.
- [185] A.K.M. Khairuzzaman, S. Chaudhury, Multilevel thresholding using grey wolf optimizer for image segmentation, *Expert Syst. Appl.* 86 (2017) 64–76.

- [186] S. Zhang, Y. Zhou, Z. Li, W. Pan, Grey wolf optimizer for unmanned combat aerial vehicle path planning, *Adv. Eng. Softw.* 99 (2016) 121–136.
- [187] Z. Boynukalin, P. Karagoz, Emotion analysis on Turkish texts, in: *Inf. Sci. Syst.* 2013, Springer, 2013: pp. 159–168.
- [188] H.G. Wallbott, K.R. Scherer, How universal and specific is emotional experience? Evidence from 27 countries on five continents, *Inf. (International Soc. Sci. Counc.* 25 (1986) 763–795.
- [189] S. Demirci, Emotion analysis on Turkish tweets, Middle East Technical University, 2014.
- [190] M.A. Tocoglu, A. Alpkocak, Emotion extraction from turkish text, *Proc. - 2014 Eur. Netw. Intell. Conf. ENIC 2014.* (2014) 130–133. doi:10.1109/ENIC.2014.17.
- [191] M.A. Tocoglu, O. Ozturkmenoglu, A. Alpkocak, Emotion analysis from Turkish tweets using deep neural networks, *IEEE Access.* 7 (2019) 183061–183069. doi:10.1109/ACCESS.2019.2960113.
- [192] F. Ren, C. Quan, Linguistic-based emotion analysis and recognition for measuring consumer satisfaction: an application of affective computing, *Inf. Technol. Manag.* 13 (2012) 321–332.
- [193] H.Y. Yen, P.H. Lin, R. Lin, Emotional product design and perceived brand emotion, *Int. J. Adv. Psychol.* 3 (2014) 59–66.
- [194] H. Davis, S.M. Mohammad, Generating music from literature, *ArXiv Prepr. ArXiv1403.2124.* (2014).
- [195] M. Hartner, The Lingering After-Effects in the Reader’s Mind An Investigation into the Affective Dimension of Literary Reading, *JLTONline Rev.* (2013).
- [196] J. Kleres, Emotions and narrative analysis: A methodological approach, *J. Theory Soc. Behav.* 41 (2011) 182–202.
- [197] S.M. Mohammad, From once upon a time to happily ever after: Tracking emotions in mail and books, in: *Decis. Support Syst.*, 2012. doi:10.1016/j.dss.2012.05.030.
- [198] X.G. Li, S.M. Li, L.R. Jiang, S. Bin Zhang, Study of english pronunciation quality evaluation system with tone and emotion analysis capabilities, in: *Appl. Mech. Mater.*, 2014: pp. 318–323.

- [199] C. Suero Montero, J. Suhonen, Emotion analysis meets learning analytics: online learner profiling beyond numerical data, in: Proc. 14th Koli Call. Int. Conf. Comput. Educ. Res., 2014: pp. 165–169.
- [200] H. Dogan, Emotion, confidence, perception and expectation case of mathematics, *Int. J. Sci. Math. Educ.* 10 (2012) 49–69.
- [201] A.D.I. Kramer, The spread of emotion via Facebook, in: Proc. SIGCHI Conf. Hum. Factors Comput. Syst., 2012: pp. 767–770.
- [202] B.-K.H. Vo, N. Collier, Twitter emotion analysis in earthquake situations, *Int. J. Comput. Linguist. Appl.* 4 (2013) 159–173.
- [203] C. Cherry, S.M. Mohammad, B. De Bruijn, Binary classifiers and latent sequence models for emotion detection in suicide notes, *Biomed. Inform. Insights.* 5 (2012) BII--S8933.
- [204] M.J. Paul, M. Dredze, You are what you tweet: Analyzing twitter for public health, in: Fifth Int. AAAI Conf. Weblogs Soc. Media, 2011.
- [205] Y. Chen, Y. Zhou, S. Zhu, H. Xu, Detecting offensive language in social media to protect adolescent online safety, in: 2012 Int. Conf. Privacy, Secur. Risk Trust 2012 Int. Confernece Soc. Comput., 2012: pp. 71–80.
- [206] M. Dadvar, D. Trieschnigg, R. Ordelman, F. de Jong, Improving cyberbullying detection with user context, in: Eur. Conf. Inf. Retr., 2013: pp. 693–696.
- [207] T. Malti, T. Krettenauer, The relation of moral emotion attributions to prosocial and antisocial behavior: A meta-analysis, *Child Dev.* 84 (2013) 397–412.
- [208] E. Grijalva, D.A. Newman, L. Tay, M.B. Donnellan, P.D. Harms, R.W. Robins, T. Yan, Gender differences in narcissism: A meta-analytic review., *Psychol. Bull.* 141 (2015) 261.
- [209] A. Minamikawa, H. Yokoyama, Personality estimation based on weblog text classification, in: Int. Conf. Ind. Eng. Other Appl. Appl. Intell. Syst., 2011: pp. 89–97.
- [210] C.S. Montero, M. Munezero, T. Kakkonen, Investigating the role of emotion-based features in author gender classification of text, in: Int. Conf. Intell. Text Process. Comput. Linguist., 2014: pp. 98–114.

- [211] D.S. Lassen, A.R. Brown, Twitter: The electoral connection?, *Soc. Sci. Comput. Rev.* 29 (2011) 419–436.
- [212] D. Maynard, A. Funk, Automatic detection of political opinions in tweets, in: *Ext. Semant. Web Conf.*, 2011: pp. 88–99.
- [213] M.D. Conover, J. Ratkiewicz, M. Francisco, B. Gonçalves, F. Menczer, A. Flammini, Political polarization on twitter, in: *Fifth Int. AAAI Conf. Weblogs Soc. Media*, 2011.
- [214] M.D. Conover, B. Gonçalves, J. Ratkiewicz, A. Flammini, F. Menczer, Predicting the political alignment of twitter users, in: *2011 IEEE Third Int. Conf. Privacy, Secur. Risk Trust 2011 IEEE Third Int. Conf. Soc. Comput.*, 2011: pp. 192–199.
- [215] A. Muddiman, N.J. Stroud, News values, cognitive biases, and partisan incivility in comment sections, *J. Commun.* 67 (2017) 586–609.
- [216] A. Ucan, B. Naderalvojud, E.A. Sezer, H. Sever, SentiWordNet for New Language: Automatic Translation Approach, in: *2016 12th Int. Conf. Signal-Image Technol. Internet-Based Syst.*, IEEE, 2016: pp. 308–315. doi:10.1109/SITIS.2016.57.
- [217] H. Rong, T. Ma, J. Cao, Y. Tian, A. Al-Dhelaan, M. Al-Rodhaan, Deep rolling: A novel emotion prediction model for a multi-participant communication context, *Inf. Sci. (Ny)*. 488 (2019) 158–180. doi:10.1016/j.ins.2019.03.023.
- [218] J. Cho, R. Pappagari, P. Kulkarni, J. Villalba, Y. Carmiel, N. Dehak, Deep Neural Networks for Emotion Recognition Combining Audio and Transcripts, in: *Interspeech 2018*, ISCA, ISCA, 2018: pp. 247–251. doi:10.21437/Interspeech.2018-2466.
- [219] X. Zhao, C. Wang, Z. Yang, Y. Zhang, X. Yuan, Online News Emotion Prediction with Bidirectional LSTM, in: 2016: pp. 238–250. doi:10.1007/978-3-319-39958-4_19.
- [220] S.M. Mohammad, S. Kiritchenko, Using Hashtags to Capture Fine Emotion Categories from Tweets, *Comput. Intell.* 31 (2015) 301–326. doi:10.1111/coin.12024.
- [221] W. Li, H. Xu, Text-based emotion classification using emotion cause extraction, *Expert Syst. Appl.* 41 (2014) 1742–1749. doi:10.1016/j.eswa.2013.08.073.

- [222] B. Desmet, V. Hoste, Emotion detection in suicide notes, *Expert Syst. Appl.* 40 (2013) 6351–6358. doi:10.1016/j.eswa.2013.05.050.
- [223] S.M. Mohammad, #Emotional Tweets, *Proc. First Jt. Conf. Lex. Comput. Semant.* 1 Proc. Main Conf. Shar. Task, Vol. 2 Proc. Sixth Int. Work. Semant. Eval. (2012) 246–255.
- [224] Y. Li, H. Su, X. Shen, W. Li, Z. Cao, S. Niu, *DailyDialog: A Manually Labelled Multi-turn Dialogue Dataset*, (2017).
- [225] CrowdFlower, (n.d.). <https://gallery.azure.ai/Experiment/Logistic-Regression-for-Text-Classification-Sentiment-Analysis-1> (accessed December 25, 2019).
- [226] A. Go, R. Bhayani, L. Huang, Twitter sentiment classification using distant supervision, *CS224N Proj. Report, Stanford*. 1 (2009) 2009.
- [227] A. B.S., N. Soni, S. Dixit, Crowdsourcing – A Step Towards Advanced Machine Learning, *Procedia Comput. Sci.* 132 (2018) 632–642. doi:10.1016/j.procs.2018.05.062.
- [228] L.A.M. Bostan, R. Klinger, An Analysis of Annotated Corpora for Emotion Classification in Text Title and Abstract in German, *Proc. 27th Int. Conf. Comput. Linguist.* (2018) 2104–2119.
- [229] T. Pires, E. Schlinger, D. Garrette, How multilingual is Multilingual BERT?, *ArXiv Prepr. ArXiv1906.01502*. (2019).
- [230] V. Sanh, L. Debut, J. Chaumond, T. Wolf, DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter, (2019) 2–6.
- [231] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Brew, HuggingFace’s Transformers: State-of-the-art Natural Language Processing, *ArXiv. abs/1910.0* (2019).
- [232] X. Li, S. Feng, D. Wang, Y. Zhang, Context-aware emotion cause analysis with multi-attention-based neural network, *Knowledge-Based Syst.* (2019). doi:10.1016/j.knosys.2019.03.008.
- [233] I. Goodfellow, Y. Bengio, A. Courville, *Deep learning*, MIT press, 2016.
- [234] J.M. Joyce, Kullback-Leibler Divergence, in: M. Lovric (Ed.), *Int. Encycl. Stat. Sci.*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011: pp. 720–722.

doi:10.1007/978-3-642-04898-2_327.

[235] T. Dozat, Incorporating nesterov momentum into adam, (2016).

EKLER

EK 1 - Tezden Türetilmiş Yayınlar

- Uçan, A., Sezer E.A., (2020). Türkçe Bilgisayarlı Dil Bilimi Çalışmalarında His Analizi. *Türk Dili Araştırmaları Yıllığı - Belleten*
- Uçan, A.,Dörterler, M., Sezer E.A., (2020). An Emotion Analysis Scheme Based on Gray Wolf Optimization and Deep Learning. *Concurrency and Computation: Practice and Experience*,(SCI-E)
- Uçan, A.,Dörterler, M., Sezer E.A., (2020). A study of Turkish Emotion Classification with Pretrained Language Models.*Journal of Information Science*,(SCI-E)
-

EK 2 - Tezden Türetilmiş Bildiriler

- Naderalvojud, B., Ucan, A., Sezer, E. A. (2018). HUMIR at IEST-2018: Lexicon-Sensitive and Left-Right Context-Sensitive BiLSTM for Implicit Emotion Recognition. *In Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis* (pp. 182-188).
- Uçan, A., & Sezer, E. A. (2019, April). A New Approach on Emotion Analogy by Using Word Embeddings. *In 2019 27th Signal Processing and Communications Applications Conference (SIU)* (pp. 1-4). IEEE.