T.C.
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

DERİN ÖĞRENME YÖNTEMLERİYLE SOSYAL MEDYA ANALİZİ VE KULLANICI TEMSİLİ

İbrahim Rıza HALLAÇ

Doktora Tezi

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

Kuramsal Temeller Bilim Dalı

TEMMUZ 2021

T.C.
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Bilgisayar Mühendisliği Anabilim Dalı

Doktora Tezi

DERİN ÖĞRENME YÖNTEMLERİYLE SOSYAL MEDYA ANALİZİ VE KULLANICI TEMSİLİ

Tez Yazarı
İbrahim Rıza HALLAÇ

Danışman
Doç. Dr. Galip AYDIN

TEMMUZ 2021
ELAZIĞ

T.C.
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Bilgisayar Mühendisliği Anabilim Dalı

Doktora Tezi

Başlığı:	Derin Öğrenme Yöntemleriyle Sosyal Medya Analizi ve Kullanıcı Temsili
Yazarı:	İbrahim Rıza HALLAÇ
İlk Teslim Tarihi:	14.06.2021
Savunma Tarihi:	12.07.2021

TEZ ONAYI

Fırat Üniversitesi Fen Bilimleri Enstitüsü tez yazım kurallarına göre hazırlanan bu tez aşağıda imzaları bulunan jüri üyeleri tarafından değerlendirilmiş ve akademik dinleyicilere açık yapılan savunma sonucunda OYBİRLİĞİ ile kabul edilmiştir.

		İmza
Danışman:	Doç. Dr. Galip AYDIN Fırat Üniversitesi, Mühendislik Fakültesi	Onayladım
Başkan:	Doç. Dr. İlhan AYDIN Fırat Üniversitesi, Mühendislik Fakültesi	Onayladım
Üye:	Doç. Dr. Mehmet Sıddık AKTAŞ Yıldız Teknik Üniversitesi, Elektrik-Elektronik Fakültesi	Onayladım
Üye:	Dr. Öğr. Üyesi Beytullah YILDIZ Atılım Üniversitesi, Mühendislik Fakültesi	Onayladım
Üye:	Dr. Öğr. Üyesi Mustafa KAYA Fırat Üniversitesi, Teknoloji Fakültesi	Onayladım

Bu tez, Enstitü Yönetim Kurulunun ...../...../20..... tarihli toplantısında tescillenmiştir.

İmza

Doç. Dr. Kürşat Esat ALYAMAÇ
Enstitü Müdürü

BEYAN

Fırat Üniversitesi Fen Bilimleri Enstitüsü tez yazım kurallarına uygun olarak hazırladığım "Derin Öğrenme Yöntemleriyle Sosyal Medya Analizi ve Kullanıcı Temsili" Başlıklı Doktora Tezimin içindeki bütün bilgilerin doğru olduğunu, bilgilerin üretilmesi ve sunulmasında bilimsel etik kurallarına uygun davrandığımı, kullandığım bütün kaynakları atıf yaparak belirttiğimi, maddi ve manevi desteği olan tüm kurum/kuruluş ve kişileri belirttiğimi, burada sunduğum veri ve bilgileri unvan almak amacıyla daha önce hiçbir şekilde kullanmadığımı beyan ederim.

12.07.2021

İbrahim Rıza HALLAÇ

ÖNSÖZ

Nesneler özellikleriyle yani kendilerini meydana getiren öge ve bileşenlerle ifade edilir. Son yıllarda gelişiminde oldukça hızlı ivme kazanan yapay zekânın çalışma alanlarından birisi de içerisinde bulunduğumuz dünyanın anlaşılması, içerisindeki varlıkların temsil edilmesidir. Yapay zekânın bir alt kolu olan derin öğrenme yöntemleri ile doğal dil işleme alanının mekanik yöntemlerden kalma yaklaşımları insanların öğrenme yeteneklerini taklit etmeye yönelik olarak rota değiştirmiştir.

Dünyada 4 milyarın üzerinde internet kullanıcısı bulunmaktadır. 2021 yılı itibariyle mikro blog servislerinden Twitter 330 milyon aktif kullanıcıya sahiptir. Kullanıcılarla ilgili hiçbir bilginin yer almadığı yalın durumda 330 milyon boyutlu bir-sıcak kodlanmış vektörlerle temsil edilmeleri söz konusudur.

Sosyal medya ağlarının yaygın kullanımıyla ortaya çıkan verilerin kullanıcı bazında yapısal temsillerinin elde edilmesi birçok bilgi getirimi (information retrival) sisteminde kıymetli uygulamaların ortaya çıkmasını sağlayacaktır. Bu tez çalışmasında güncel olan en iyi metin temsil yöntemlerinden yararlanılarak sosyal medya kullanıcılarının temsillerinin elde edilmesi için bir yöntem önerilmiştir. Ayrıca, derin öğrenme yöntemleriyle Türkçe sosyal medya verileri üzerinde duygu analizi ve metin sınıflandırma problemlerinin çözümünde başarıyı arttıran yöntemler önerilmiştir.

"Sosyal Ağlar İçin Bulut Tabanlı, Dağıtık İtibar Analizi Sistemi" projemi 1512 başlıklı programı çerçevesinde destekleyen TÜBİTAK'a, "Derin Öğrenme Tabanlı ve Yüksek Doğruluklu Yüz Tanıma ile Geçiş Yetkilendirme ve Takip Sistemi" projemi AR-GE ve İnovasyon Destek Programı çerçevesinde destekleyen KOSGEB'e teşekkürlerimi borç bilirim.

Sosyal medya analizi ve metin madenciliği faaliyetlerini gerçekleştirmek üzere araştırmacı olarak görev aldığım "Derin Öğrenme Büyük Veri Analizi Platformu (Değirmen)" projesi ekip arkadaşlarıma ve projeyi destekleyen Savunma Sanayii Başkanlığı'na teşekkürlerimi borç bilirim.

Doktora çalışmam süresince, değerli görüş ve katkılarıyla beni yönlendiren, her konuda önerileri ile desteğini esirgemeyen tez danışmanım Sayın Doç. Dr. Galip AYDIN'a teşekkürü borç bilirim.

Son olarak sadece tez döneminde değil, her zaman yanımda olup bana destek olan değerli aileme teşekkür ederim. Tezimi, dürüstlüğü ve çalışkanlığıyla her zaman örnek aldığım ve kendisini kaybetmiş olmanın üzüntüsünü yaşadığım canım babam Ömer HALLAÇ'a ithaf ederim.

İbrahim Rıza HALLAÇ
ELAZIĞ, 2021

İÇİNDEKİLER

Sayfa

ÖZET

Derin Öğrenme Yöntemleriyle Sosyal Medya Analizi ve Kullanıcı Temsili

İbrahim Rıza HALLAÇ

Doktora Tezi

FIRAT ÜNİVERSİTESİ
Fen Bilimleri Enstitüsü

Bilgisayar Mühendisliği Anabilim Dalı

Temmuz 2021, Sayfa: xiii + 138

Eğitim, alışveriş, eğlence gibi hayatın önemli bir bölümünün çevrimiçi dünyaya taşındığı günümüzde her an devasa miktarlarda veri üretilmektedir. Sadece metin, ses ve görüntülerin yalın formatlarının değil, çevrimiçi dünyada yer alan kişilerin (sosyal medya profili, yazar, eğitimci, öğrenci, vb.), ürünlerin (giysi, kitap, film, şarkı) ve daha birçok nesnenin semantik temsillerinin elde edilmesi amacıyla geliştirilecek yöntemlerin makine öğrenmesinin yakın geleceğinde önemli bir yer bulacağı kuvvetle muhtemeldir. Bu tez çalışmasında, Twitter kullanıcılarının profil temsilleri elde edilerek sosyal medya analizi uygulamalarında geniş bir yelpazede kullanım alanı bulacak bir yöntem geliştirilmiştir. Güncel en iyi durumun sunduğu derin öğrenme tabanlı metin temsil yöntemlerinden yararlanılarak, yapısal ve yapısal olmayan kullanıcı bilgilerinden bir arada yararlanabilen denetimsiz bir kullanıcı temsil öğrenme modeli önerilmiştir. Önerilen yöntemin gerçekleştirilmesi ve başarı testlerinde kullanılmak üzere, semantik olarak gruplara ayrılmış kullanıcılara ait profil bilgileri, paylaşımları, yorumları ve beğenileri gibi birçok bilgiden oluşan türünün tek örneği bir sosyal medya veri seti oluşturulmuştur. Önerilen kullanıcı temsil modelinin elde edilmesinde tf-idf, word2vec, doc2vec, ELMO, BERT gibi birçok metin temsil yönteminin farklı şekillerde kullanılması sonucunda elde edilen sonuçlar detaylı bir şekilde karşılaştırılmıştır. Özellikle, dağıtık doküman temsili modelleri için kapsamlı bir parametre incelemesi yapılmıştır.

Kullanıcı temsillerinin elde edilmesine ek olarak sosyal medya ve diğer internet kullanıcılarının oluşturdukları içerikler üzerinde derin öğrenme yöntemleriyle metin sınıflandırma ve duygu analizi uygulamalarında karşılaşılan iki önemli probleme yönelik önerilen yaklaşımlarla literatüre katkı sağlanmıştır. Duygu analizi probleminde, farklı kaynaklardan elde edilen zengin bir Türkçe veri seti oluşturularak bu veri setinin dengesiz olmasından kaynaklanan ve derin öğrenme modelinin başarısını sınırlayan koşulların aşırı örnekleme yöntemleriyle aşılmasına yönelik kapsamlı bir çalışma gerçekleştirilmiştir. Metin sınıflandırma probleminde ise, az miktarda etiketli tweetin olduğu koşullarda yüksek doğrulukta sınıflandırma yapan, derin öğrenme modelleri üzerinde uygulanabilecek bir alan adaptasyonu yaklaşımı önerilmiştir. Çalışmada kullanılmak üzere 5 sınıf için etiketlenmiş Türkçe tweet veri seti oluşturulmuştur. Önerilen yaklaşımlar CNN, LSTM gibi klasik derin öğrenme yöntemleriyle birlikte, bir arada kullanıldıkları hibrit mimariler tasarlanarak kapsamlı bir inceleme yapılmıştır.

Anahtar Kelimeler: Kullanıcı temsili, Derin öğrenme, Doğal dil işleme

ABSTRACT

Social Media Analysis and User Representation with Deep Learning Methods

İbrahim Rıza HALLAÇ

Ph.D. Thesis

FIRAT UNIVERSITY
Graduate School of Natural and Applied Sciences

Department of Computer Engineering

July 2021, Pages: xiii + 138

In today's world, where a significant part of life such as education, shopping and entertainment has shifted to the online world, huge amounts of data are being produced at every moment. It is very likely that besides the representation of simple text, sound and images, studies on semantic representation of things such as people (social media profile, author, educator, student, etc.), products (clothes, books, movies, songs) and many other objects in the online world will find an important place in the near future of machine learning. In this thesis, we propose a method for obtaining profile representations of Twitter users that will find a wide range of use cases in social media analysis applications. Our method, based on state-of-the-art deep learning-based text representation methods, can learn from both structured and unstructured user data in an unsupervised manner. We created a one-of-a-kind social media dataset to implement and evaluate the models. The dataset consists of rich user information such as profile information (biography, location, etc.), shares (retweets), comments and likes from semantically grouped users. The proposed user representation models were compared in detail for the different scenarios of exploring different text representation techniques such as tf-idf, word2vec, doc2vec, ELMO, BERT. In particular, a comprehensive parameter review has been performed for distributed document representation models.

In addition to obtaining user representations, another contribution has been made by proposing approaches to two important problems encountered in text classification and sentiment analysis applications using deep learning methods on the text content created by social media and other Internet users. In the sentiment analysis problem, a Turkish dataset was created from multiple sources and a comprehensive study was conducted using the oversampling technique to overcome the imbalanced class distribution problem that limits the success of deep learning models. In the case of the text classification problem, a domain adaptation approach is proposed that can be applied to deep learning models that classify with high accuracy under conditions with a small amount of labelled tweets. For use in the study, a Turkish tweet dataset was created and labelled for 5 classes. For both studies, a comprehensive investigation of the proposed approaches was conducted using classical deep learning methods such as CNN, LSTM and hybrid architectures where they are used together.

Keywords: User representation, Deep learning, Natural language processing

ŞEKİLLER LİSTESİ

Sayfa

TABLOLAR LİSTESİ

Sayfa

SİMGELER VE KISALTMALAR

Simgeler

α_i	: Veri setindeki i. eğitim kullanıcısı
β_j	: Veri setindeki j. test kullanıcısı
d	: Vektör boyutu
d_{jk}	: j. test kullanıcısı ve k. grubun temsillerinin Öklid uzaklığı
$\mathbb{E}_{\alpha}^{i}$	: i. eğitim kullanıcısının gömmesi
$\mathbb{E}_{\beta}^{j}$	: j. test kullanıcısının gömmesi
$\mathbb{E}_{\theta}^{k}$	: k. grubun ortalama grup gömmesi
M	: Eğitim kullanıcısı sayısı
N	: Test kullanıcısı sayısı
θ_k	: k. grup
S	: Kullanıcı veri setindeki grup sayısı
$\vec{T}$	: Tweet gömmesi/temsili/vektörü (embedding)
V	: Sözlükte yer alan farklı kelime sayısı
w_i	: Sözlüğün i. kelimesi

Kısaltmalar

AÖF	: Aşırı Örnekleme Sayısı
BERT	: Bidirectional Encoder Representations from Transformers
BiLSTM	: İki Yönlü Uzun/Kısa Süreli Bellek (Bidirectional Long/Short-Term Memory)
CBOW	: Sürekli Kelime Torbaları (Continuous Bag of Words)
CNN	: Evrişimli Sinir Ağı (Convolutional Neural Network)
DBOW	: Dağıtık Kelime Torbaları (Distributed. Bag of Words)
DDİ	: Doğal Dil İşleme
ELMO	: Embeddings from Language Models
FFNN	: İleri Beslemeli Sinir Ağı (Feedforward Neural Network)
IDF	: Ters Doküman Frekansı (Inverse Document Frequency)
İDM	: İstatistiksel Dil Modeli
MLP	: Çok Katmanlı Algılayıcı (Multi Layer Perceptron)
NLD–RLC	: Name Location Description – Retweet Like Comment
LSTM	: Uzun Kısa/Süreli Bellek (Long/Short Term Memory)
LR	: Öğrenme Oranı (Learning Rate)
OOV	: Sözlük Dışı (Out-of-Vocabulary)
PV-DM	: Paragraph Vector Distributed Memory
PWE	: Önceden-Eğitilmiş Kelime Gömmeleri (Pre-Trained Word Embeddings)
RNN	: Tekrarlayan Sinir Ağları (Recurrent Neural Networks – RNN)
SGD	: Stokastik Gradyan İniş (Stochastic Gradient Descent)
SVD	: Tekil Değer Ayrışımı (Singular Value Decomposition)
TF	: Terim Frekansı
t-SNE	: t-Distributed Stochastic Neighbor Embedding
YSA	: Yapay Sinir Ağı

1. GİRİŞ

20. yüzyılın başlarında dile ait kuralların belirlenip mekanik veya elektronik bir sistem ile otomatikleştirilerek bir dilden diğerine makine çevirisi yapılacağı öngörülmüştü. İlerleyen süreçte, doğal dillerin kural tabanlı modelleme teknikleriyle ve zamanın hesaplama ortamı koşullarında insana yakın çeviri yapılmasına imkân vermeyecek derecede zengin olduğu gerçeğiyle karşılaşılmıştır. Son yıllarda, makine öğrenmesi ve doğal dil işleme (DDİ) alanlarında devrim niteliğinde yeni teknikler ortaya çıkmıştır. Lojistik Regresyon, Naive Bayes, Destek Vektör Makineleri gibi yöntemler artık geleneksel makine öğrenmesi yöntemleri olarak anılmaktadır. TF-IDF, kelime çantası gibi metin temsil yöntemleri de "geleneksel", "klasik" gibi ifadelerle anılmaktadır. Bu yöntemleri demode yapan en zayıf yönleri, özelliklerin elle seçiliyor olması ve kelime frekansı gibi önceden tanımlı, modelin insan gibi düşünmesine elverişli olmayan yapıda olmalarıdır.

Kelimelerin ve dokümanların semantik uzayda temsillerinin elde edilmesi konusunda son yıllarda heyecan verici gelişmeler olmuştur ve her geçen gün yeni teknikler ortaya çıkmaktadır. Bu yöntemlerin başarısı, kelime temsilleri arasında "kraliçe − kadın + erkek ≈ kral" ve "İstanbul − Türkiye + Londra ≈ İngiltere" gibi analojilerin Öklid uzaklığı, kosinüs benzerliği gibi benzerlik ölçütleri üzerinden görülebilmektedir. Hatta gerçek hayatta var olan cinsiyetçilik, ırkçılık gibi çeşitli sorunlardan kaynaklı önyargı ve meyil (bias) içeren değerlendirmelerin bile semantik uzayda kelime temsillerine aktarıldığı görülmektedir [1]. Bir kelime temsil yönteminin başarısını değerlendirmenin bir diğer yolu ise onu kümeleme, sınıflandırma gibi DDİ problemlerinde diğer temsil yöntemleriyle kıyaslamaktır (task oriented evaluation).

Sosyal medya kullanıcılarının da tıpkı kelimeler gibi semantik uzayda temsil edilmesi ve bu temsiller arasında doğruluk ölçütü olarak çeşitli analojilerin yürütülmesi mümkündür. Tezde ortaya konulan yaklaşımlar ve sonuçların değerlendirilmesi için bazı varsayımlarda bulunulmuştur:

- Moda dünyasından bir kullanıcı temsilinin semantik uzayda spor dünyasından bir kişiye kıyasla moda dünyasından başka bir kişiye daha yakın olması gerekir.
- Daha derine indiğimizde demokrat bir politikacının temsilinin cumhuriyetçi kullanıcıların temsillerine kıyasla demokrat kullanıcılara daha benzer olmasını bekleriz.

Veri setinde yer alan elementler arasındaki benzerlik ve farklılıkların incelenmesi temel veri madenciliği problemlerinden biridir [2]. Dijital platformlardaki kullanıcılar arasındaki benzerliklerin tespit edilmesi konusunda önemli çalışmalar gerçekleştirilmiştir [3–5]. Bu çalışmalarda önerilen yaklaşımların ortak hedefi; kullanıcıların ve veri içerisinde yer alan diğer elementlerin, bulunan mecradaki özelliklerine ve birbirleriyle olan ilişkilerine paralel olarak

vektörel temsillerinin elde edilmesidir. Sadece ürettiği içeriklerden yola çıkarak bir sosyal medya kullanıcısının ilgi alanları, siyasi görüşü, ruhsal durumu vb. hakkında fikir sahibi olabiliriz. Sosyal medya profilinin ortaya koyacağı bu özellikler ve daha fazlasının tespit edilmesinde diğer kullanıcıların paylaşımları üzerinden gerçekleştirdiği etkileşimler (beğeni, yorum vb.), takip ettiği kişiler, onu takip edenler, konumu, profil resmi, kendisi hakkında yazdığı kısa açıklama ve daha birçok yapısal olmayan bilgiden yararlanılabilir. Görüleceği üzere, bir sosyal ağda kullanıcı temsilleri elde ederken yararlanılabilecek oldukça fazla çeşit ve miktarda veri mevcuttur. Hem miktar hem de tür olarak bütün verilerden yararlanabilen algoritmalar geliştirmek oldukça zor bir problemdir. Öncelikle olası bütün verilere eksiksiz olarak üçüncü kişilerin erişmesi yasal açıdan mümkün olmamakla birlikte erişim yetkisi verilen bilgilerin kullanımı konusunda kullanıcıların kişilik haklarını korumak üzere belirlenmiş geniş kapsamlı kısıtlamalar mevcuttur [6]. Yasal açıdan bir engel olmadığı takdirde bile teknolojik yeterlilik bakımından bu devasa verinin depolanması ve işlenmesi için gerekli donanımsal ve yazılımsal altyapı ile bu süreçlerin yönetimini ve yürütülmesini sağlayacak insan kaynağına sahip olma gücü ancak Facebook, Twitter, Google gibi şirketler için söz konusudur. Dolayısıyla kullanıcı temsillerinin elde edilmesi problemi penceresinden bakıldığında, herkese açık (public) verilerden olabildiğince yararlanabilen yöntemlerin geliştirilmesine dair yaklaşımlar öne sürmenin önemi ortaya çıkmaktadır.

Twitter örneğinde, bir kullanıcı profili manuel olarak analiz edildiğinde, ilk aşamada; temel kullanıcı bilgileri olan isim, açıklama (description), konum ve profil resmi ile birlikte takip edilenler (followees), takip edenler (followers) ve güncel paylaşımlar (tweet, retweet) etkili olacaktır. Ardından, kullanıcıyla etkileşimde bulunan diğer kişiler üzerinden çeşitli çıkarımlar yapılabilir. Bu etkileşimlerden kasıt, analiz edilen kullanıcının paylaşımlarını kimlerin beğendiği (like), yeniden paylaştığı (retweet), altına yorum yazdığı (comment) gibi aktivitelerdir. Bir insan tarafından birkaç dakika gibi kısa bir sürede gerçekleştirilebilecek bu incelemenin makine öğrenmesi yöntemleriyle insansız olarak gerçekleştirilebilmesi için bahsedilen geniş kapsamlı verileri otomatik olarak işleyen ve anlamlı çıkarımlar yapabilen algoritmalara ihtiyaç vardır. Bu tezin ana amaçlarından biri kullanıcılarla ilgili farklı türde bilgilerden üretilebilen temsillerin elde edilmesidir.

Bu tez çalışması 7 temel bölümden oluşmaktadır. Tezin yapısı aşağıdaki gibi düzenlenmiştir:

Bölüm 1 Giriş bölümüdür. Bu bölümde tezin amacı, kapsamı ve içeriği özetlenmiştir.

Bölüm 2'de diğer bölümlerde önerilen yöntemlerin temelini oluşturan DDİ tekniklerinin anlatımına yer verilmiştir. Eski yerel temsil yöntemlerinden (kelime çantası, n-gram vb.) bu günlerde hemen hemen her DDİ probleminde yararlanılmakta olan dağıtık temsiller ve bağlama

duyarlı modellere kadar tez kapsamında gerçekleştirilen çalışmalarda kullanılan bütün tekniklerin temel çalışma prensipleri anlatılmıştır.

Bölüm 3'te dengesiz veri seti üzerinde gerçekleştirilen duygu analizi çalışması yer almaktadır. Türkiye'deki çeşitli popüler web sitelerinden (Beyazperde, N11, Tripadvisor ve Kitapyurdu) elde edilen duygu veri seti üzerinde gerçekleştirilen bu çalışmada derin öğrenme yöntemlerinin uygulanması öncesinde gerçekleştirilen kapsamlı veri analizi ve veri görselleştirme aşamaları detaylı olarak sunulmuştur. Sınıf dengesizliği problemine dikkat çekilerek aşırı örnekleme frekansı adını verdiğimiz veri çoğaltma miktarının belirlendiği bir parametre üzerinden farklı derin öğrenme modellerinin performansları incelenmiştir. Kullanılan klasik derin öğrenme modelleri (MLP, CNN, LSTM) ve önerilen hibrit modeller (BiLSTM-CNN, CNN-BiLSTM) için gerekli bilgiler verilmiştir.

Bölüm 4'te kültür, ekonomi, politika, spor veya teknoloji alanları için az miktarda etiketli tweetin mevcut olduğu koşullarda derin öğrenme modellerinin yüksek doğrulukla metin sınıflandırma görevini gerçekleştirmeleri amacıyla önerilen alan adaptasyonu (domain adaptation) yöntemi sunulmuştur.

Bölüm 5'te, sosyal medya kullanıcılarını temsil etmek için, doc2vec dağıtık doküman temsili tekniği kullanılarak bir nöral kullanıcı temsili (neural user embedding) modeli önerilmiştir. Aynı zamanda Bölüm 6'da önerilen gelişmiş kullanıcı temsil yönteminin de temelini oluşturan bu çalışmada, 500 Twitter kullanıcısına ait 500.000 tweetten oluşan altın standart bir kullanıcı veri setinin elde edilme adımları anlatılmıştır. Farklı metin ön işleme adımları uygulanarak önerilen modelin en başarılı olduğu ön işleme yöntemi araştırılmıştır. Yöntemin başarısının farklı parametrelere (iterasyon sayısı, vektör boyutu) göre değişimini gösteren deneysel çalışmalar sunulmuştur.

Bölüm 6'da önerilen yaklaşım geliştirilerek sadece tweet metinlerinden değil, tweetler üzerinden gerçekleştirilen etkileşim bilgileri ile elde edilen bilgilerden de faydalanarak daha başarılı temsiller üreten yeni bir model önerilmiştir. Çalışmada kullanılmak üzere 2 yeni Twitter kullanıcı veri seti oluşturulmuştur. Geleneksel yöntemler (TFIDF), klasik kelime gömmeleri (word2vec vb.), doc2vec algoritmaları ve bağlamsal metin temsil yöntemleri (BERT vb.) üzerinden ayrıntılı bir karşılaştırmaya yer yerilmiştir. Literatürde öne çıkan yöntemlerin ve bu yöntemlerin başarısına katkı sağlama potansiyeli olan farklı türden veri kaynaklarının kullanıldığı deneylerin sonuçları sunulmuştur.

Bölüm 7 Sonuçlar bölümüdür. Her bölüm sonunda gerçekleştirilen çalışmanın sonuç ve katkılarına detaylı olarak yer verildiği için bu bölümde sadece önerilen yöntemlerin genel bir değerlendirmesi yapılmıştır. Ayrıca, gelecek çalışmalara yönelik önerilerde bulunulmuştur.

2. METİN TEMSİL YÖNTEMLERİ

Makine öğrenmesinde modellenecek parametrelere ait değişkenlerin belirlenebilmesi için öncelikle veriye ait özelliklerin belirlenmesi gerekir. Örneğin balık türlerinin otomatik olarak tahmin edildiği bir görevde iki boyutlu fotoğrafların piksel değerleri öznitelik olarak seçilebilir. Bir diğer alternatif ise uzunluk sensörleri yardımıyla elde edilen en ve boy uzunluklarının öznitelik olarak kullanılmasıdır.

Metin; karakter, kelime, cümle, paragraf gibi çeşitli alt birimlerden oluşur. Metin madenciliğinde herhangi bir doğal dilde yazılmış olan metin onu sayısal dünyada temsil eden, nümerik değerlerden oluşan bir vektörle temsil edilir. Metin içerisinde bulunan hangi özelliklerin ne şekilde kullanılacağı konusunda birçok DDİ tekniği mevcuttur.

Bu bölümde bilgisayarlı dilbilim alanının başlangıcından günümüze doğru gelişim süreci ele alınarak tezin diğer bölümlerinde kullanılan DDİ teknikleri için gerekli temel bilgilere yer verilmiştir.

2.1. Doğal Dil İşleme ve Metin Temsil Yöntemlerinin Kısa Geçmişi

Doğal dil, insanların temel iletişim aracıdır. Konuşmalar ve yazılı metinler esas itibariyle bilgi ve düşünce barındırır. Aklımızdan geçen düşüncelerin zihnimizden başka bir insanın zihnine aktarılması sesler (söylenen sözcükler), resimler, şekiller gibi bir temsil ve kural sistemi üzerinden yapılır. Örneğin, iletişim kuran iki insan arasında transfer edilen doğal dil birimleri birbiri ardına gelen semboller kümesi olarak modellenebilir [7]. İdeal bir sistemin bu aktarım sırasında herhangi bir bilgi veya düşünce kaybına ihtimal vermemesi beklenir. Sümerli rahiplerin bundan yaklaşık 5500 yıl öncesinde tapınaklarına getirilen mahsul miktarlarını kil tabletlere kaydetmesi [8], üniversitelerin öğrencilerine ait isim, yaş, bölüm, memleket gibi bilgileri veri tabanında saklaması ya da günümüzde bir sosyal ağ içerisindeki kullanıcıların diğer hangi kullanıcılarla doğrudan bağlantılı olduğunun çizge olarak ifade edilmesinde nicelik olarak bir bilgi kaybı söz konusu değildir. İfade edilen bilgi ve düşünce karmaşıklaştıkça onu temsil eden dile ait özelliklerin de karmaşıklaşması doğaldır. Doğal dil işlemenin mekanik hesaplamalarla başlayıp özellik mühendisliği yaklaşımlarıyla devam eden serüveninde egemen olarak makine öğrenmesi yöntemlerinden yararlandığımız bir aşamada bulunmaktayız.

Bilgisayarlara doğal dillerde yazılı metinleri anlama becerisi kazandırarak metin özetleme, soru cevaplama, tercüme etme, bilgi getirimi (information retrival) ve bilgi çıkarımı (information extraction) gibi görevleri insanlar için yerine getirebilecek sistemlerin üretilmesi alanında yapılan çalışmalar gün geçtikte önemini arttırmaktadır. Bu çalışmalar bilgisayar bilimi, dilbilim (linguistics) ve yapay zekâ alanlarının kesişimi olan doğal dil işleme alanının konusudur.

Bilgisayarlı dilbilimde yaşanan gelişmeler incelendiğinde ele alınan konuların genellikle çözülmesi gereken yeni problemlerin ortaya çıkmasına neden olduğu görülmektedir. Bazen asıl problemlere göre daha çözülebilir olduğu ya da ticari olarak daha fazla değer verildiği için ortaya çıkan bu yeni problemlere daha fazla ilgi gösterilmiştir [9].

Veri madenciliği ile normalde veri içerisindeki varlığı aşikâr olmayan fakat muhtemelen yararlı olabilecek her türlü bilgi veya örüntünün keşfedilmesi amaçlanır. Veri madenciliği yöntemleri yapısal (structured) veya yapısal olmayan (unstructured) farklı tür, büyüklük, çoğalma hızına sahip verilere uygulanır. Oldukça büyük ve çeşitli miktarda metin verisinin mevcut olduğu günümüzde bu verilerin toplanması, saklanması ve işlenmesinde kullanılan tekniklerde de ilerlemeler kaydedilmiştir. Dolayısıyla veriden bilgiye dönüşüm olanaklarında iyi yönde muazzam bir değişim söz konusudur. Metin verisinin yapısal olmaması münasebetiyle öncelikle metinden sayısal dünyaya dönüşümün yani metinlerin nasıl temsil edileceğinin belirlenmesi gerekmektedir. Metin temsil yöntemlerinin geçmişten bugüne nasıl evirildiğini anlayabilmek için DDİ, yapay zekâ ve hesaplamalı dilbilim (computational linguistics) alanlarındaki önemli süreçlerin incelenmesinde yarar vardır.

Dilin bir sistem olarak ele alınması ve hesaplanabilir problemler içeren bir bilim dalı olarak kabul görmesi yönündeki ilk gelişmeler 20. yüzyıl başlarında gerçekleşmiştir [10]. Daha sonraki süreçte makine çevirisi konusunda ilk çalışmaların ortaya çıktığı görülmektedir. Makine çevirisi çalışmaları doğal dil işleme alanının da başlangıcı kabul edilmektedir. Rus mucit Petr Petrovich Troyanskii 1933 yılında mekanik olarak çeviri yapan bir cihaz için yaptığı patent başvurusu ile bilinen ilk makine çevirisi çalışmalarından birini gerçekleştirmiştir [11]. Önerilen yöntemde sözlüklerin mekanikleştirilmesi ve kısıtlı miktarda gramer kuralının uygulanması söz konusudur. 1946 sonrasında W. Weaver, A.D. Booth ve makine çevirisi alanındaki diğer öncü bilim adamlarının yayınladıkları raporlar, gerçekleştirdiği toplantılar ve önerdiği projelerle elektronik bilgisayarlar yardımıyla çevirinin nasıl yapılabileceği konusundaki fikirlerin ortaya çıktığı görülmektedir [12]. Tıpkı mikroskobun icadının gözle görülemeyen hücrelerin görüntülenmesine imkân vererek biyoloji bilimini değiştirmesi gibi bilgisayarların da doğal dilleri derinden analiz ederek dilbilimine yön verme potansiyeli söz konusudur [13].

Bilgi getirimi alanı için aynı dönemlerde benzer gelişmeler meydana gelmiştir [14]. Örneğin, 1920 ve 1930'lu yıllarda Emanuel Goldberg tarafından alınan çeşitli patentlerde mekanik olarak mikrofilm şeridi üzerinde arama yapmak yoluyla katalog taraması benzeri işlevleri gerçekleştiren sistemlerden bahsedilmektedir. Bilgi erişimi alanında bilgisayarın kullanılmasına yönelik ilk örneklerden birisi ise 1948'de bir konferansta Holmstrom tarafından açıklanan bir doküman arama sistemidir. Bu sistem manyetik teyp üzerinde dokümanları önceden belirli sembollerle ilişkili olarak kaydedebilen ve bu semboller üzerinden arama yaparak dokümanları getirebilen Univac adı verilen bir makinedir.

1954 yılında Georgetown Üniversitesi ve IBM iş birliğiyle bilgisayarların sadece fizik, kimya gibi temel bilimlerin sayısal problemlerini çözen cihazlar olarak kalmayıp doğal dil işleme problemlerinin çözümünde de kullanılmasının en önemli adımlardan birisi atılmıştır. IBM 701 adı verilen bilimsel araştırmalara yönelik ilk IBM bilgisayarı kullanılarak 60'tan fazla Rusça cümlenin İngilizceye elektronik olarak çevrildiği bir gösterim yapılmıştır. Georgetown-IBM deneyi adı verilen otomatik çeviri sistemi prototipi 250 kelimelik bir elektronik sözlük ve 6 adet gramer kuralı içermektedir. O tarihlerde kaydedilen bu aşamada henüz bilgisayarlı dilbilimde günümüzdeki kelime temsil yöntemlerine (örneğin kelime frekans dizileri) benzer bir metin temsil yönteminin kullanılmadığı görülmektedir. Georgetown deneyinde kullanılan 700 serisi makinede standart IBM delikli kart (punch card) bulunmaktadır [15]. Mimari açıdan oldukça basit olan bu sistemde tercüme edilecek Rusça cümleler önce kelime kelime okunmaktadır. Sonrasında kelimelerde yer alan karakterlerin ikili (binary) ASCII karşılıkları hafızada aranmaktadır. Cümlelerin istatiksel veya herhangi bir matematiksel işleme tabi tutulmasını sağlamak üzere ortaya konulmuş bir metin temsil yöntemi söz konusu değildir. Boşluk karakterlerinin tespiti ile cümlelerin ayrıştırılması (tokenization) işlemi gerçekleştirilerek elde edilen kelimelerin İngilizcedeki karşılıkları elektrostatik hafızada önceden tanımlı yerleştirme kurallarıyla birlikte kodlanmıştır. Hafıza erişim teknikleri üzerinden üretilen çözümlerle kelime arama modülü ve tercüme kurallarının uygulanması gerçekleştirilmiştir. Arama yönteminde; başlangıç harfine göre hafızada her kelime için belirli bir alan indekslenmektedir ve aranan kelimeler içerisinde en fazla eşleşen karaktere sahip kelime bulunmaktadır.

Dilbilimi ve yapay zekâ alanındaki gelişmeler incelendiğinde 1950'li yılların bu dönemde ortaya çıkan fikirler ve yöntemler nedeniyle önemli bir zaman dilimi olduğu görülmektedir. Deterministik problemleri çözmek için bilgisayarların nasıl programlanması gerektiği konusunda çalışmaların devam ettiği bu yıllarda, insan gibi düşünebilen genel amaçlı sistemlerin nasıl programlanabileceği sorusuna cevap arayan önemli adımlar atılmış ve ilk kez yapay zekâ kavramından bahsedilmiştir [16,17]. Programlama dillerini bilgisayarlarla aramızda bir köprü olarak kullanmaya daha iyi bir alternatif şüphesiz doğal diller üzerinden anlaşmaktır. McCarthy, programlama dillerinin gelişigüzel görevleri yerine getirecek şekilde kullanılabilmesi için bütün davranışların sistem içerisinde temsil edilebilir olması gerektiğini ortaya koymuştur [18]. Chomsky ise 1957 yılındaki sözdizimsel yapılar (syntactic structures) çalışmasında ele aldığı semantik, sözdizim, derin yapı, yüzey yapı kavramlarıyla dilbilim dünyasına yeni bir bakış açısı kazandırmıştır [19]. Chomsky, dil içerisinde gramer yapısı olarak doğru fakat bir anlamı olmayan cümlelere dikkat çekmiş ve bilgisayarın doğal dilleri anlaması için cümle yapısının belirli bir şekilde değiştirilmesi gerektiğini öne sürmüştür. Ayrıştırma (parsing) işlemi ile cümleler dilin sözdizim (syntax) kurallarına göre parçalara ayrılır. Olasılıksal üretici dil modelinde yapı olarak bir dilde var olması mümkün olan her cümlenin gramer kurallarına uyması gerekir. Bir cümleyi

anlamaya çalıştığımızda sadece gramer bilgimiz değil o güne kadar sahip olmuş olduğumuz bütün bilgi birikimi, zekâ, kelime bilgisi, cümlenin bağlamı gibi farklı kavramlar devreye girer. Bundan dolayı doğal dilin bilgisayar tarafından modellenebilmesi ve anlaşılması ancak bu kavramların bir arada programlanabilmesiyle mümkündür [20].

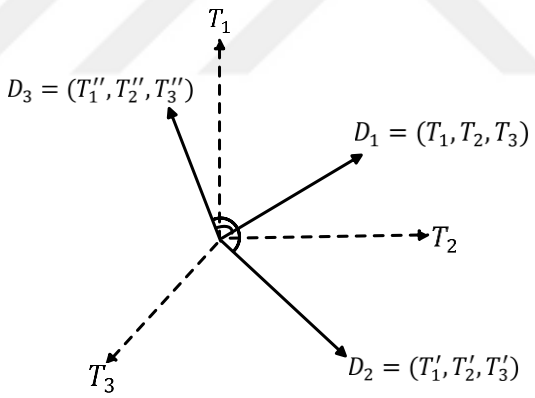
Doğal dillerin matematiksel olarak nasıl modelleneceği ve makine çevirisi gibi problemlerin çözümünde nasıl başarılı olacağı konusunda ortaya atılan fikirler çok güçlü teoriler barındırsa da oldukça yüksek beklentiler ortaya koymuş ve bir süre sonra (1970'lere doğru) bu alanda gerçekleştirilen çalışmalar durma noktasına gelmiştir. Bunun en önemli sebeplerinden birisi makine çevirisi konusunda verilen büyük maddi desteklere rağmen beklenen başarının elde edilememesidir. Makineler, çeviri konusunda insanlarla yarışamayacak kadar yavaş ve kötü çeviriler üretmekteydi. Makinelerin insanlardan ayırt edilemeyecek derecede inandırıcı bir şekilde insanlarla iletişim kurabileceği yönünde ortaya atılan Turing testi için gelinen seviye de yine pek umut verici değildi.

Metin verisinin özelliklerinin sayısal değerlerle ifade edilmesinde kullanılan yöntemler kronolojik olarak incelendiğinde ise başlangıçta günümüzdeki şekliyle kelime veya doküman temsillerinden bahsedilmediği anlaşılmaktadır. 1940'lı yılların sonlarına doğru, başta bilgi erişimi ve makine çevirisi alanları olmak üzere, mekanik sistemlerle gerçekleştirilen kural tabanlı çözümlerin bu kez bilgisayarlarla kelime kelime okunan metinler üzerinde uygulanması söz konusudur. Bununla birlikte hafızada saklanan sözlük yapıları ve bu sözlüklerde yer alan kelimelerin sayısal değerlerle (indeksleme) ifade edildiği, fakat dönemin bilgisayarlarının hafıza ve işlem kapasitesi bakımından oldukça kısıtlı olması sebebiyle bu işlemlerin oldukça sınırlı büyüklükteki sözlüklerle yapıldığı çalışmalar görülmektedir. Ayrıca, başta metin sınıflandırma çalışmalarında olmak üzere, dokümanlar önceden belirlenen anahtar kelimeleri içerip içermemelerine göre bir-sıcak kodlama (one-hot encoding) benzeri mantıksal vektörler ve matrislerle temsil edilmiştir [21,22]. Aslında anahtar kelimelerle doküman temsillerinin çıkış noktası ilk kez Luhn [23] tarafından önerilen TF (term frequency) yöntemidir. TF ile dokümanlar içerdiği kelimelerin frekans değerleriyle ifade edilmektedir. Bu yöntemin farklı bir versiyonu olarak, dokümanları belirleyici özel olarak seçilmiş kelimelerin listesi oluşturulup bu kelimelerin bulunması (0/1) veya kaç kez bulunduğu (frekansı) bilgisi kullanılabilir. Doğal dil işlemenin bu zaman dilimi (1940 - 1960) için 1954 ve 1957 yıllarına ait iki çalışma istatistiksel semantik alanının temelini oluşturmaları ve modern kelime gömme [24] (word embedding) yöntemlerinin ilham kaynağı olmaları nedeniyle dikkat çekmektedir. Bunlardan birincisi aynı zamanda kelime torbası (bag of words) konseptinin ilk kez bahsedildiği Zellig Harris'e ait Dağılımsal Yapı (Distributional Structure) [25] çalışmasıdır. Firth'e ait çalışmada [26] ortaya konulan hipoteze göre kelimeler anlamsal olarak ilişkili olduğu diğer kelimelerle aynı pencerelerde yer alır. Bu hipotez "You shall know a word by the company it keeps" sözleriyle açıklanmıştır. "Bana

arkadaşını söyle, sana kim olduğunu söyleyeyim" atasözündeki insanları tanıma konusunda izlenebilecek yaklaşıma benzer şekilde kelimeleri de bir arada bulunduğu diğer kelimelerden yola çıkarak tanıyabiliriz.

Sparck Jones 1970'li yılların başında TF yöntemini geliştirmek üzere getirdiği ağırlıklandırma önerisiyle kelimelerin dokümanlara özel olmalarını hesaba katabilecek daha gelişmiş bir doküman temsil yaklaşımına sahip TF-IDF (term frequency – inverse document frequency): "terim frekansı - ters doküman frekansı" yöntemini ortaya koymuştur [27]. Dokümanların terim sayılarının birbirinden farklı olmalarından kaynaklanacak dezavantajlı koşulların üstesinden gelebilmek için TF ve IDF değerlerini farklı şekillerde normalize eden çeşitli TF-IDF türevleri bulunmaktadır ve yeni versiyonları geliştirilmeye devam edilmektedir [28,29].

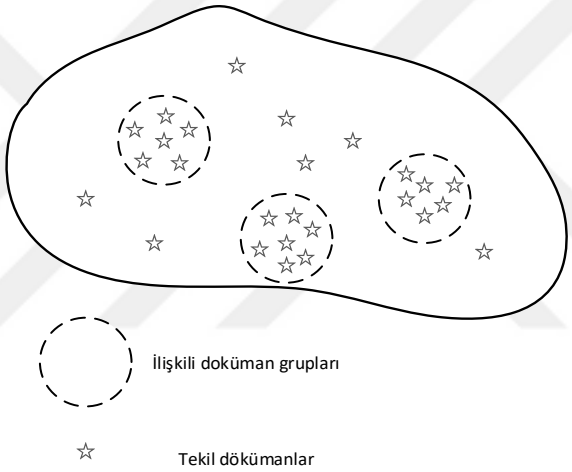
1975 yılında Salton tarafından önerilen vektör uzayı modeli (vector space model) [30] ile dokümanların birbirlerine benzerliklerini yansıtabilen bir uzayda temsil edilmeleri gösterilmiştir. Şekil 2.1 örnek bir doküman vektör uzayını göstermektedir. D ile gösterilen D_i dokümanlarının içerdiği t_1, t_2, t_3 terimlerinin ağırlık değerlerine göre vektörel büyüklükleri birbirine göre çizildiğinde aralarındaki açının büyüklüğüyle ters orantılı olarak bir benzerlik göstermektedir.



Şekil 2.1. Vektör uzayında doküman gösterimi [30]

Vektörler iki boyutlu uzayda nokta şeklinde belirtildiğinde ise bu benzerlik değerleri noktaların birbirlerine olan yakınlarından gözlenebilir. Şekil 2.2, kendi içinde benzerlik gösteren doküman kümelerini ve bu kümelerin birbirinden nasıl ayrıldığını göstermektedir. Bu modelin getirdiği en önemli avantajlardan birisi benzer anlamlar ifade eden fakat farklı terimlerden oluşan cümleleri ve aynı şekilde ortak terimlerden oluşmasına rağmen farklı anlamlar ifade cümleleri ayırabilmesidir. Örneğin "hasret hisseden insanların fikirleri değerlidir", "özlem duyan kişilerin düşünceleri kıymetlidir" cümlelerinde olduğu gibi terimler farklı olsa da ağırlık değerleri birbirine yakın olacağı için anlam benzerliği tespit edilir.

Vektör uzay modelinde dokümanlar içerdikleri semantik özellikleriyle kodlanması yaklaşımıyla birlikte doküman vektörleri üzerinden konuların temsil edilmesine olanak sağlayan LSI (latent semantic idexing) [31] gibi yöntemlerin gelişmeye başladığı görülmektedir (1985 ve sonrası). LSI, insan hafızasının modellenmesi gibi daha geniş bir alanda problemlere uygulanarak daha kapsayıcı bir isim olarak literatürde LSA (latent semantic analysis) [32] olarak geçmektedir. LSA, doküman temsillerinde saklı (latent) semantik özellikleri elde etmek için doküman vektörlerinden oluşan matris üzerinde SVD (singular value decomposition) uygulayarak daha küçük boyutlu, konuları temsil eden vektörler elde eder. Konular, dokümanların bütünü, yani derlem (corpus) içerisinde aranır ve bilgi getirimi gibi DDİ uygulamalarında dokümanlarla olan benzerlik ilişkileri üzerinden çeşitli analizler gerçekleştirilir. Örneğin Şekil 2.1'de gösterilen doküman vektör uzayı için Tablo 2.1'de verilen terim-doküman matrisi elde edilecektir.



Şekil 2.2. İdeal doküman vektör uzayı [30]

Tablo 2.1. Doküman vektörleri

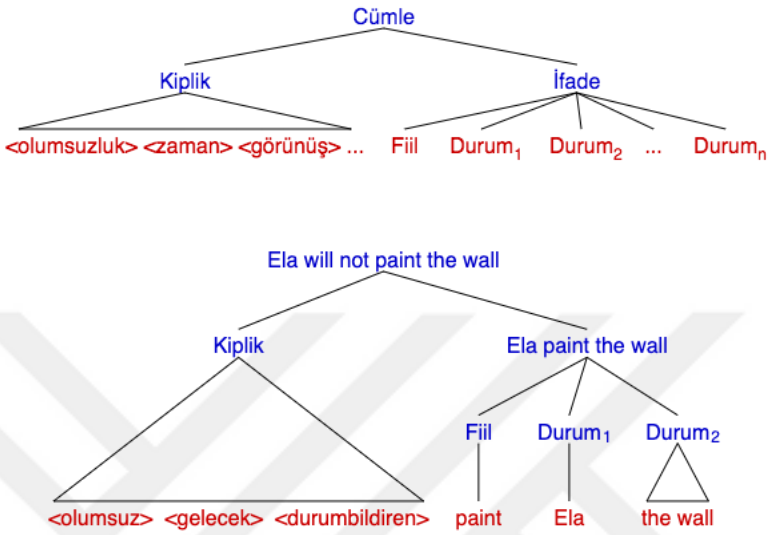
	$\boldsymbol{D_1}$	$\boldsymbol{D_2}$	$\boldsymbol{D_3}$
$\boldsymbol{t_1}$	T_1	T_2	T_2
$\boldsymbol{t_2}$	T_1'	T_2'	T_3'
$\boldsymbol{t_3}$	T_1''	T_2''	T_3''

Doğal dil modelleme ve temsil yöntemleri bakımından önemli bir diğer problem aynı zamanda bilgi gösterimi (knowledge representation) alanının da konusu olan metinlerin gerçek dünyada sahip olduğu bilgi, anlam, sözdizim ve gramer yapısı gibi özelliklerin temsil edilmesi kapsamında gerçekleştirilen çalışmalardır. Gramer kuralları, dilin üretebileceği cümleleri belirler ve türetilen cümlelerin anlamları kısmen içerdiği terimlerin sözdizimleriyle kodlanmış olur [33]. İngilizce için çözümlenmiş bir durum dilbilgisi (case grammar) temsili örneği Şekil 2.3'te "Ela

will not paint the wall" cümlesi üzerinden ağaç veri yapısı ile gösterilmiştir. Ayrıştırma kuralı şu şekildedir:

$$Cümle \rightarrow Kiplik\ (modality) + İfade\ (proposition)$$

$$İfade \rightarrow\ Durum_1 + Durum_2 + Durum_3 + \cdots + Durum_n$$



Şekil 2.3. Cümlelerin sözdizim ağacıyla temsil edilmesi [34]

Anlam bilgisini yapısal bir formda göstermek için yararlanılan bir diğer yöntem semantik ağ oluşturmaktır. Semantik ağlarda ana fikir; bir dilin benzer anlamlı kelimelerinin tanımlanan bir veri yapısı üzerinden ilişkilendirilmesidir. Kelimeler çizge (graph) veya ağ yapılarında olduğu gibi düğümleri, aralarındaki eş anlamlılık gibi ilişkiler ise bağlantıları oluşturur. 1990'lı yıllarda gerçekleştirilen WordNet [35] çalışmasında kelimeler isim, fiil, sıfat ve zarf olarak sınıflara (syntactic category) ayrılıp aralarındaki alt anlamlılık (hyponymy), eş anlamlılık (synonymy), zıt anlamlılık (antonymy), parça-bütün (meronymy) ilişkileri tanımlanmıştır. Bu süreçlerdeki karar mekanizması dil bilimi konusunda yeterli bilgi birikimine sahip uzmanlardır. Daha sonraki yıllarda (2004) Türkçe için bir sözcük ağı geliştirilmiştir [36] ve benzer çalışmalar devam etmiştir.

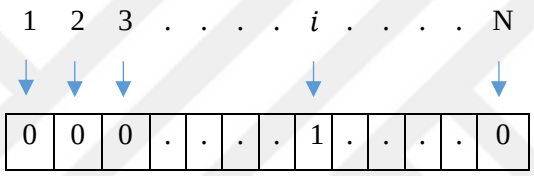
2.2. Eski ve Temel Yöntemler

Mevcut metin temsil yöntemleri, üç ana grup altında ele alınabilir: klasik kelime temsilleri, karakter düzeyinde özellikler ve bağlamsal kelime temsilleri [37]. Kelime torbası, n-gramlar ve bir-sıcak kodlama gibi yöntemler yerel temsiller olarak tanımlanmaktadır. Derin öğrenme tabanlı yeni DDİ yöntemleri dağıtık temsiller gibi sürekli (continuous) temsiller ile daha

etkin başarı göstermektedir. Tez çalışmasında önerilen yöntemlerin eski, temel yaklaşımlarla karşılaştırıldığı durumlar söz konusu olduğu için bu yöntemlerin anlatımına kısaca yer verilmiştir.

2.2.1. Bir-Sıcak Kodlama

İsimlendirmesini sayısal devrelerde bir mantıksal değerin sıfır veya birlerden oluşmasından alan bu kodlama sistemi aynı zamanda makine öğrenmesinde kategorik verilerin gösteriminde kullanılmaktadır. Bir metin içerisindeki kelimelerin kodlanması olarak bir-sıcak kodlama (one-hot encoding) kelimelerin N-boyutlu vektörler halinde temsil edilirken sadece bir pozisyondaki değerin "1" diğerlerinin "0" olduğu, oldukça basit bir yöntemdir. Bir kelimeye ait bir-sıcak kodlama örneği Şekil 2.4'te gösterilmiştir. Bu yöntemde cümleler temsil edilirken içerdikleri bir-sıcak kodlanmış vektörler alt alta getirilerek matris şeklinde temsil edilir.



Şekil 2.4. Bir-sıcak kodlama

Metin içerisinde geçen bütün farklı kelimelerin oluşturduğu kümeye sözlük denilmektedir. Bin farklı kelimeden oluşan bir metin ($N = 1000$) içerisindeki i. farklı kelime için "1", diğer 999 değer için sıfır olan $\mathbb{R}^{|N| \, x \, 1}$ vektörüyle gösterilir. $F(i) = 1$ şeklinde formüle edilebilen bu gösteriminde kelimeler ayrık olarak temsil edilir ve bağlam bilgisi kodlanmaz.

Bir-sıcak kodlama yönteminde vektör boyutunun çok büyük olmasının önüne ikili-kodlama (binary encoding) yöntemiyle geçilebilir. Kelimenin sözlük içerisinde aldığı tamsayı değerinin ikili sayı sistemindeki karşılığı kullanıldığında vektör uzunluğu N'den çok daha küçük olan log_2^N olur.

2.2.2. Kelime Torbası

Standart kelime frekansı gösterimi olan kelime torbası (bag-of-words) yaklaşımı ismini bir çanta veya torba içerisindeki nesnelerin toplanırken geliş sıralarını kaybetmesi benzetmesinden alır. Dokümanda yer alan kelimelerin kaç adet olduğu bilinir fakat hangilerinin yan yana geçtiği bilgisi kaybedilir. Bu yaklaşımda sözlük oluşturmak için derlemdeki bütün kelimeler taranır ve farklı-terim-sayısı boyutunda doküman vektörleri elde edilir.

x: Yunkai tahtında bet yüzlü bir kraliçe vardı ve kraliçe mutluydu.

y: Braavos tahtında temiz yüzlü bir kral ve huzurlu bir kraliçe vardı.

z: Halk huzurlu ve mutluydu.

Örneğin x, y ve z dokümanlarının kelime torbası yöntemiyle vektörleri elde edilirken 14 farklı kelime içeren bir sözlük meydana gelir. Kelimeler rasgele, frekanslarına göre veya alfabetik olarak numaralandırılabilir.

Tablo 2.2'de verilen kelimeler; 1: "ve", 2: "kraliçe"; 3: "bir", 4: "tahtında", 5: "mutluydu" , 6: "vardı", 7: "huzurlu", 8: "yüzlü", 9: "temiz", 10: "kral", 11: "halk", 12: "yunkai", 13: "bet", 14: "braavos" şeklinde numaralandırılmıştır.

Tablo 2.2. Kelime torbası yaklaşımı

Doküman	Vektör													
x	1	2	1	1	1	1	0	1	0	0	0	1	1	0
y	1	1	2	1	0	1	1	1	1	1	0	0	0	1
z	1	0	0	0	1	0	1	0	0	0	1	0	0	0

Dokümanların her üçünde "ve" bağlacı birer kez geçtiği için doküman vektörlerinin ilk boyutundaki değerleri üçünde de 1'dir. Görüleceği üzere doküman vektörlerinde çok sayıda 0 değeri bulunmaktadır, yani oldukça seyrek (sparse) vektörler oluşturulmuştur. Diğer bir problem ise gerçek hayat problemlerinde sözlükte bulunan farklı kelime sayısı bu örnekte olduğu gibi 14 değil binlerce katı daha büyük bir değer olacaktır. Boyutun bu derece büyük olması istenen bir durum değildir.

2.2.3. Terim Frekansı - Ters Doküman Frekansı (TF-IDF)

Frekans tabanlı en başarılı doküman temsil yöntemi olarak kabul edilen TF-IDF 1972 yılında Sparck Jones tarafından önerilmiştir. Dokümanların terim frekans vektörleri (TF) olarak temsil edilirken her kelime için elde edilecek ters doküman frekansı adı verilen (IDF) değerleriyle çarpılması fikrine dayanmaktadır. N adet dokümandan oluşan D derlemi içerisindeki i. dokümanının (D_i), j. terimine ait TF-IDF ağırlığı şu şekilde hesaplanır:

$$TF - IDF\ (D_{ij}) = \mathrm{TF}(D_{ij}) * \log\left(\frac{N}{Df_j}\right) \tag{2.1}$$

Df_j değeri sözlükteki j. terimin tüm derlem içerisindeki toplam bulunma sayısını göstermektedir. Formüldeki TF değeri ise normalize edilmiş terim frekansıdır. Bunun için terim frekansı dokümandaki toplam terim sayısına (Df_i) bölünür. Literatürde Df_i değerine alternatif olarak derlemde en çok bulunan terimin toplam frekansı veya toplam terim sayısı gibi farklı değerlerin de tercih edildiği görülmektedir.

Örnek olarak kelime torbası yöntemi gösteriminde kullanılan x, y, z dokümanlarından oluşan derlemdeki dokümanların TF-IDF vektörlerini hesaplarsak, öncelikle diğer frekans hesabına dayalı tekniklerde olduğu gibi sözlük oluşturulur. Tablo 2.3'te gösterildiği şekilde sözlükteki terimlerin her doküman için ayrı ayrı TF değerleri ve tüm dokümanlarda ortak olarak kullanılan IDF değerleri hesaplanır. Her bir dokümanın TF-IDF vektörleri Tablo 2.4'te verilmiştir.

Tablo 2.3. TF ve IDF değerleri

Keliemeler (j)	Df_{xj}	Df_{yj}	Df_{zj}	Df_j	IDF	**TF(x)**	**TF(y)**	**TF(z)**
ve	1	1	1	3	0	.1	.091	.25
kraliçe	2	1	0	3	0	.2	.091	0
bir	1	2	0	3	0	.1	.182	0
tahtında	1	1	0	2	.176	.1	.091	0
mutluydu	1	0	1	2	.176	.1	.000	.25
vardı	1	1	0	2	.176	.1	.091	0
huzurlu	0	1	1	2	.176	0	.091	.25
yüzlü	1	1	0	2	.176	.1	.091	0
temiz	0	1	0	1	.477	0	.091	0
kral	0	1	0	1	.477	0	.091	0
halk	0	0	1	1	.477	0	0	.25
yunkai	1	0	0	1	.477	.1	0	0
bet	1	0	0	1	.477	.1	0	0
braavos	0	1	0	1	.477	0	.091	0

Tablo 2.4. TF-IDF vektörleri

Doküman	TF-IDF Vektörü													
x	0	0	0	.018	.018	.018	0	.018	0	0	0	.048	.048	0
y	0	0	0	.016	0	.016	.016	.016	.043	.043	0	0	0	.043
z	0	0	0	0	.044	0	.044	0	0	0	.119	0	0	0

Bir terim bütün dokümanlarda bulunduğunda IDF değeri log(1) yani sıfır olur. Bu şekilde bir kelimenin IDF değeri o dokümana özel olması oranında yüksek olur. Aynı düşünceyle, çoğu dokümanda yer alan, doküman hakkında bilgi vermeyen kelimeler düşük IDF değerine sahip olur. TF ve IDF değerlerinden herhangi birinin sıfır olması TF-IDF değerini sıfır yapmaya yeteceğinden dolayı elde edilen vektörde kelimenin o dokümanda hiç yer almadığı durum ile her dokümanda bulunuyor olması durumu aynı şekilde kodlanmış olur. Literatürde yer alan TF-IDF formülü versiyonlarının bazılarında IDF değerine her zaman 1 gibi sabit bir sayı eklenerek bu

durumun ortadan kaldırıldığı görülmektedir. Kelime torbası, bir sıcak kodlama ve TF-IDF yöntemlerinin genel bir karşılaştırılması Tablo 2.5'te yer almaktadır.

Tablo 2.5. Frekans hesabına dayalı yöntemlerin karşılaştırılması

Yöntem	**Örnek Vektör**	**Derlemdeki bilgilerin kullanılma seviyesi**	**Vektör Boyutu**
Bir sıcak kodlama	[0 0 1 0 0 …. 0]	Sözlükte yer alan kelimelerin listesi kullanılır.	Sözlükte bulunan terim sayısı
Kelime torbası	[2 1 5 0 3 …. 4]		
TF-IDF	[0.021 0.044 0 0.031 …. 0]	Terimlerin hem temsil edilen dokümana hem de tüm derleme ait frekans değeri kullanılır.	

Bir sıcak kodlama ve kelime torbası yöntemlerinde temsil edilen doküman dışındaki dokümanlara ait terim frekans bilgileri kullanılmaz. Bu özellikleri ile TF-IDF yönteminden farklıdırlar.

2.2.4. N-Gram

Bir metin içerisinde kelimelerin veya daha küçük birimlerin (karakter, hece, gövde) n tanesinin birbiri ardına gelerek oluşturduğu yapıya n-gram denir ve n = 1,2,3 … 1-gram, 2-gram, 3-gram şeklinde belirtilir. Dörtten küçük n-gramların özel isimleri bulunmaktadır. Bunlar sırasıyla unigram, bigram ve trigram şeklindedir. Örneğin "yapay sinir ağlarının temel özellikleri" cümlesi için n-gramlar Tablo 2.6'daki gibidir.

Tablo 2.6. N-gram ile özellik çıkarımı

1-gram	yapay	sinir	ağlarının	temel	özellikleri
2-gram	yapay sinir	sinir ağlarının	ağlarının temel	temel özellikleri	
3-gram	yapay sinir ağlarının	sinir ağlarının temel	ağlarının temel özellikleri		
4-gram	yapay sinir ağlarının temel	sinir ağlarının temel özellikleri			
5-gram	yapay sinir ağlarının temel özellikleri				

N-gramlar, kelime torbası ve TF-IDF yöntemlerinden farklı olarak kelimelerin hangi sırada bulunduğu bilgisini kaybetmez. "sinir ağı", "temel özellik" gibi bigramların ve "yapay sinir ağı" gibi trigramların derlemde farklı dokümanlarda yeniden bulunması mümkündür ve bu n-gramların keşfedilmesi ile metin temsili zenginleşecektir. N-gramların bağlam yönünden

getirdiği bu avantaj dört veya daha büyük n-gramlar için azalmaya başlar, çünkü bu kelime grupları derlem içerisinde oldukça seyrek bulunur. Giriş verisinin özellikleri göz önünde bulundurularak vektör boyutu büyüklüğü, eklenen yeni boyutların kazandırdığı bilgi oranına göre anlamlı bir seviyede tutulur. Ayrıca, N-gramlar sözlük içerisine dahil edilerek "n-gram torbaları" (bag-of-n-grams) gibi genişletilmiş özellik çıkartım teknikleri uygulanabilir.

2.2.5. Özellik Vektörü

Dokümanları çeşitli dilsel özellikleriyle temsil etme yöntemidir. Bu özellikler kelime sayısı, karakter sayısı gibi doğrudan elde edilebilen bilgiler olabileceği gibi sözcük türü, varlık ismi gibi elde edilmesi için öncelikle çeşitli yöntem ve araçlara ihtiyaç duyan özellikler de olabilir. Vektörlerde yer alabilecek bilgiler çok çeşitli olup genel olarak aşağıda verildiği gibi 4 farklı gruba ayrılabilir:

- Kelime, karakter, rakam, özel karakter sayıları vb.
- Varlık ismi tanıma (named entity recognition) yoluyla kişi, yer, kurum vb.
- Sözcük türü işaretleme yoluyla (part-of-speech tagging) isim, sıfat, fiil, zarf vb.
- Semantik ağlar yoluyla alt anlamlılık, eş anlamlılık, zıt anlamlılık vb.

Bu bilgilerden yola çıkarak elde edilebilecek örnek bir özellik vektörü Tablo 2.7'de verilmiştir.

Tablo 2.7. Özellik vektörü yaklaşımı

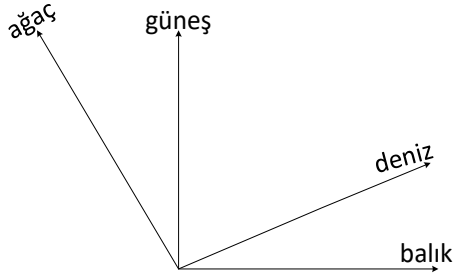
Doküman	**Vektör**				
	kelime sayısı	**yer adı sayısı**	**kişi adı sayısı**	**fiil sayısı**	**…**
x	23	2	1	2	…

Tablo 2.8. Birliktelik değerleri

Kelime-1	**Kelime-2**	**Birliktelik Frekansı**
balık	deniz	87
deniz	güneş	56
güneş	balık	24
ağaç	güneş	67
balık	ağaç	8

Sayma tabanlı özellik vektörü elde etme yöntemlerinden bir diğeri de kelime birliktelik (co-occurrence) frekanslarıdır. Kelimelerin bir arada bulunma sayılarını elde etmek için cümle, doküman, n-gram gibi bir kapsama alanı seçilerek derlem içerisinde bu alanlarda kaç kez bir arada geçtiği üzerinden bir benzerlik matrisi oluşturulur. Örneğin Wettler ve Rapp [38] birliktelik

frekansıyla elde edilen semantik kelime temsillerini benzer kelimelerin tespitinde kullanmış ve bu yöntemin başarısını insanlara "bu kelimeyi duyduğunda başka hangi kelimeler aklına geliyor/bu kelime hangi kelimeleri çağrıştırıyor" sorusunun sorulmasıyla elde edilen cevaplarla karşılaştırmıştır.



Şekil 2.5. İki boyutlu kelime vektörleri

Amasyalı ve Beken, Türkçe kelimeler için anlamsal semantik benzerlik temsilleri elde ederek bu temsillerin metin sınıflandırma probleminde TF temsillerinin kullanılmasını göre daha iyi performans sergilediğini göstermiştir [39]. Tablo 2.8'de örnek olması amacıyla hayali bir Türkçe derlem için kelime birliktelik frekansları verilmiştir. Şekil 2.5'te bu değerlerin iki boyutlu uzayda kelime temsilinde kullanmasıyla elde edilen vektörler gösterilmiştir.

2.3. Dil Modelleme

Kelimelerin hangi sıralamada bir arada olduğu bilgisinden yola çıkarak bir dizinin dil içerisinde var olma olasılığını hesaplayan yöntem veya algoritmalara dil modeli denir. Bir dizi kelimeden sonra hangi olasılıkla hangi kelimenin geleceğinin tahmin edilmesi yine dil modellemenin konusudur. Örneğin Türkçe doğal dili için geliştirilen bir konuşma tanıma (speech recognition) sistemi için eğitilmiş bir dil modelinin hedef söz öbeklerinin olasılığını tahmin ederken aşağıdakine benzer sonuçlar elde etmesi beklenir:

$$P(\text{"}ben\ de\ özledim\ ben\ de\ resmin\ var\ şu\ an\ elimde\text{"}) \approx 0.01$$

$$P(\text{"}ben\ de\ özledim\ sen\ de\ resmin\ var\ şu\ alemimde\ \text{"}) \approx 0.001$$

Bu bölümde istatistiksel dil modelleri ve nöral dil modelleri hakkında temel bilgiler verilmiştir.

2.3.1. İstatistiksel Dil Modelleri

İstatistiksel dil modeli (İDM) bir derlem içerisindeki dilsel birimlerin (kelime, cümle, doküman) bir arada bulunma koşulları (dizilim olasılıkları) ve öncül olasılık dağılımlarının belirlenmesine dayanmaktadır. DDİ alanının en önemli çalışma konularından birisi olan İDM

alanındaki gelişmeler başta konuşma tanıma [31] sistemleri olmak üzere istatistiksel makine çevirisi (statistical machine translation) [32], doğal dil üretimi (natural language generation), bilgi getirimi [33] gibi diğer alanlara doğrudan katkı sağlamaktadır.

Genel olarak dil modellemede kelimelerin soldan sağa ardışıl birimler olarak ele alınması söz konusudur. Modelleme için derlemde yer alan cümle, kelime, harf, hece gibi alt birimlerden birisi seçilmelidir. Bu birimler küçüldükçe modellenecek veri sayısı doğru orantılı olarak büyüyecektir. Doğal dillerde çok fazla farklı kelime bulunduğundan dolayı İDM'nin eğitim verisi olarak kullandığı metinler üzerinde öğrenmesi gereken çok fazla parametre söz konusu olur. Özellikle Türkçede dilin sondan eklemeli yapısı sebebiyle bir kelimeden çok fazla yeni kelime türetilebilmektedir. Örneğin *ağaç* kelimesinden türetilen *ağaç+lan+dır+ıl+ama+ma+sı+nda+ki* kelimesinin derlemde bulunma olasılığı düşük de olsa mümkündür. Bir kelimenin çok farklı ve nadir bulunan versiyonlarının sözlükte bulunması ise öğrenmeyi zorlaştırmaktadır [40]. Bu durumun diğer bir zorlaştırıcı özelliği de dil modeli eğitildikten sonra sözlükte yer almayan kelimelerle karşılaşılma olasılığının yüksek olmasıdır.

Dil modeli ile n tane kelimeden $(w_1, w_2, \dots, w_n)$ oluşan bir metnin bir cümle oluşturup oluşturamayacağının olasılığı $P(S)$ bulunabilir. Bu kestirime bir cümlenin olasılığı denir. Örneğin birinci kelime $w_1 =$ "sosyal" ise P($w_2 =$ "bilgiler"), P($w_2 =$ "sorumluluk"), P($w_2 =$ "medya"), P($w_2 =$ "güvenlik") olasılıkları P($w_2 =$ "bardağın"), P($w_2 =$ "çanta"), P($w_2 =$ "rüya") gibi kelimelerin olasılıklarından büyüktür. Çünkü Türkçe dilinde "sosyal medya", "sosyal güvenlik" gibi ikililerin sayısı "sosyal bardağın", "sosyal çanta" gibi ikililerden daha fazladır. Cümlenin olasılığı $P(S) = P(w_1, w_2, \dots, w_n)$ ilk kelimeden itibaren her kelimenin kendisinden önceki kelimelere bağımlı olarak gelmesi koşuluna göre Denklem 2.2'deki şekilde hesaplanır.

$$P(S) = P(w_1)\, x \prod_{i=2}^{n} P(w_i \mid w_{i-1}) \tag{2.2}$$

$$= P(w_1)\, P(w_2 \mid w_1)\;\; P(w_3 \mid w_1, w_2) \dots\; P(w_n \mid w_1, w_2, w_3 \dots\; w_{n-1})$$

Olasılık teorisinin zincir kuralından yola çıkarak birbirine bağlı koşullu olasılıkların yan yana çarpımıyla tek bir olasılık değeri bulunur. Dil modeli oluştururken bu şekilde ilk kelimeden son kelimeye kadar bütün kelimelerin birbiri ardına gelme durumlarının hesaplanması derlemdeki sözlük boyutunun (V) büyük olduğu problemler için uygulanması zor bir yoldur.

Markov varsayımında bir olayın gerçekleşmesi sadece bir önceki olaya bağlıdır. Yani başlangıçtan itibaren tüm olayların ele alınması devre dışı bırakılır. Markov stokastik sürecindeki bir sonraki olayın tahmin edilmesi kavramının istatistiksel dil modelleme problemindeki karşılığı; bir dizi kelimeden sonra gelecek kelimenin tahmin edilmesidir. Bu yaklaşımdan yola çıkarak bir sonraki kelimenin tahmin edilmesinde başlangıçtan itibaren bütün kelimeler ele alınmadan, sadece son N-1 tane kelimeye koşullanması anlamına gelen **N-gram** dil modelleri ortaya

çıkmıştır. N-gramların doğal dil işleme alanında metinlerin ikili, üçlü gibi alt kümelere ayrılarak bu alt kümeler üzerinden temsil edilmesi amacıyla da oldukça kullanıldığı metin temsil yönteminde gösterilmiştir. Denklem 2.3'de sadece bir önceki kelime ele alındığı için N-1 değeri 1'e eşittir. Dolayısıyla genel dil modeli yaklaşımı aynı zamanda N=2 olduğundan dolayı bir bigram dil modelidir. Markov modelinde buna birinci dereceden Markov zinciri denir.

Normalde $P(w_n \mid w_1, w_2, w_3, \dots, w_{n-1})$ olarak hesaplanacak n elemanlı bir zincir 4-gram modeline göre $P(w_n \mid w_{n-3},\ w_{n-2}, w_{n-1})$ ile hesaplanır. Yani,

$$P(S) = \prod_{i=1}^{n} P(w_i \mid w_{i-N-1}) \tag{2.3}$$

$P(w_4 \mid w_1, w_2, w_3)\ P(w_5 \mid w_2, w_3, w_4)$ ... $P(w_n \mid w_{n-3},\ w_{n-2}, w_{n-1})$ olur.

Zincir kuralıyla bütün olasılıkları birbiriyle çarpmak yerine Markov kabulüyle $P(w_n \mid w_1, w_2, w_3 \dots\ w_{n-1}) \approx P\left(w_n \mid w_{n-(N-1)}, \dots\ w_{n-1}\right)$ sadece son terimin olasılık değeriyle sonuç elde edilir. Bir kelime silsilesinde $P(w_i \mid w_{i-1})$ koşullu olasılığını $[w_1, \dots\ w_{i-2}]$ aralığındaki kelimelerden bağımsız olarak hesaplamak için kullanılabilecek en yaygın yöntemlerden birisi **Maksimum Olabilirlik Çıkarımı** (MLE) yöntemidir. Bu yöntemde $f1 \rightarrow frekans\ (w_i, w_{i-1})$ ve $f2 \rightarrow frekans\ (w_{i-1})$ ile $f1/f2$ şeklinde hesaplanır.

Tablo 2.9'da verilen örnek derlem üzerinde trigramlar üzerinden dil modeli olasılık hesabı örneği:

$$P(\text{"meşgul"} \mid \text{"böyle zamanlarda"}) = \frac{f1}{f2}$$

$$\frac{f1}{f2} = \frac{\text{"böyle zamanlarda meşgul" trigram sayısı}}{\text{"böyle zamanlarda ..." sayısı}} = \frac{2}{5} = 0.4$$

Tablo 2.9. Dil modeli örnek derlemi

1	**Böyle zamanlarda *memleketimi*** özlerim.
2	**Böyle zamanlarda *meşgul*** olman oldukça anlaşılır bir durum.
3	Şiirin anlamı kadar biçimi de önemlidir.
4	İhracat artışının **böyle zamanlarda *da*** olması mümkün.
5	Kişilerin **böyle zamanlarda *ebeveynlerine*** bakması gerekir.
6	Sizi **böyle zamanlarda *meşgul*** etmesinin bir anlamı yok.

İDM ile metin üretimi, imla hatalarının düzeltilmesi (spelling correction) gibi uygulamalar geliştirildiğinde ilk kelimeden cümlenin sonuna kadar tüm kelimeler sırayla tahmin edilir. Olasılık tahmini her yeni kelime eklendikten sonraki durum için yeniden yapılır. Bu

uygulamalar için dikkat edilmesi gereken koşullardan birisi de ilk/son kelimelerin olasılıklarının hesaplanmasıdır. Normalde bigram dil modelinde cümle olasılığı Denklem 2.4'teki gibidir.

$$P(S) = P(w_1)\,P(w_2 \mid w_1) \;\ldots\; P(w_n \mid w_{n-1}) \tag{2.4}$$

Bu formüle göre ilk kelimenin w_1 olma olasılığı tüm diğer kelimelerle eşit şekilde dağıtılırsa $P(w_1) = w_1/V$ olur. Fakat bu olasılık tam olarak doğru kabul edilemez. Çünki bu olasılık gerçekte derlemde bu kelime ile başlayan cümlelerin sayısı üzerinden hesaplanmalıdır. Bu olasılık $(w_1 \mid başlangıç)$ olarak hesaplanmalıdır. Aynı şekilde bitiş kelimeleri için olasılık $(w_n \mid bitiş)$ şeklinde kendi içerisinde hesaplanıp dil modeli içerisine dahil edilmelidir. Tablo 2.9'da verilen derleme göre $P("böyle" \mid başlangıç) = 2/6$ olur. Çoğu uygulamada başlangıç terimi <sos>, bitiş terimi <eos> olarak belirtilir. Bu belirteçler İngilizcedeki "start-of-sentence" ve "end-of-sentence" terimlerinin kısaltmalarıdır.

İstatistiksel dil modelleme yöntemlerinin önemli bir dezavantajı bulunmaktadır. Eğitim verisi olarak kullanılan derlemde yer almayan n-gramlar test derleminde mevcut değilse olasılıkları sıfır olarak tahmin edilir. Bu durumdaki kelimelere görünmeyen kelimeler denir. Hedef dil içerisindeki bütün kombinasyonların eğitim verisinde yer alması hem gerçekçi değildir hem de işlem kabiliyeti açısından çok büyük boyutlarda temsil vektörlerinin kullanılmasına neden olur. Modelin doğru çalışması için bir yandan daha fazla kelime içeren zengin bir derlemle eğitilmesi gerekirken diğer yandan özellik uzayının üssel olarak artması söz konusudur. Bu durum çok değişkenli makine öğrenmesi problemlerinde genel olarak yüksek boyut veya boyutluluk laneti (curse of dimentionality) olarak bilinmektedir. N-gram dil modellerinin görünmeyen kelimeler ve yüksek boyutluluk kaynaklı dezavantajlarını azaltmak amacıyla çeşitli yumuşatma (smooting) teknikleri önerilmiştir. Bu tekniklerden bazılarının isimleri: Kneser-Ney, Katz, Good-Turing, Jelinek-Mercer, back-off ve additive yumuşatmadır.

Tablo 2.10. Dil modeli örnek derlemi

P(w | <sos>)		**P(w | Ne)**		**P(w | geceler)**		**P(w | ne)**		**P(w | gündüzler)**		**P(w |gördüm)**	
Bir	0.25	kadar	0.32	var	0.28	hayatlar	0.18	var	0.56	var	0.40
Bu	0.15	taraf	0.18	gördüm	0.21	zamanlar	0.17	zaman	0.19	zaman	0.31
Hayat	0.09	katkı	0.12	gündüzler	0.18	**gündüzler**	**0.16**	**gördüm**	**0.18**	**<eos>**	**0.27**
Zaman	0.07	…	…	kadar	0.15	…	…	…	…	…	…
…	…	**geceler**	**0.09**	**ne**	**0.12**	…	…	…	…	…	…
Ne	**0.04**	…	…	…	…	…	…	…	…	…	…
…	…	…	…	…	…	…	…	…	…	…	…

Dil modellerinde performans ölçütü olarak genellikle çapraşıklık (perplexity) metriği kullanılmaktadır [41]. Bigram bir dil modelinin W test derlemi üzerinden başarısı her bir w_i olasılığının yan yana çarpımının kelime sayısına (N) göre normalize edilmesiyle bulunur:

$$PP(W) = \sqrt[N]{\frac{1}{\prod_{i=1}^{N} P(w_i \mid w_{i-1})}} \tag{2.5}$$

Bu formül ters olasılık değeri hesapladığından dolayı çapraşıklık değerinin düşük olması dil modelinin daha iyi olduğunu gösterir. Çapraşıklık, test derlemi için hesaplandığında modelin genel bir başarım değeri elde edilir. Tablo 2.10'da $PP(Ne\ geceler\ ne\ g\ddot{u}nd\ddot{u}zler\ g\ddot{o}rd\ddot{u}m)$ test cümlesinin başlangıç teriminden bitiş terimine kadar her kelime için bigram olasılıkları verilmiştir. Her durum için kelime olasılıklarının toplamı 1'dir.

Verilen test cümlesi 5 kelimeden oluştuğu için <sos> ve <eos> terimleriyle birlikte toplam kelime sayısı 7'dir. Dolayısıyla çapraşıklık formülünde $N = 7$ olur.

$$PP(W) = \sqrt[7]{\frac{1}{P(w_1)P(w_2)P(w_3)P(w_4)P(w_5)P(w_6)P(w_7)}} \tag{2.6}$$

$$= \sqrt[7]{\frac{1}{0.04\ x\ 0.09\ x\ 0.12\ x\ 0.16\ x\ 0.18\ x\ 0.27}}$$

Sonuç olarak $PP(Ne\ geceler\ ne\ g\ddot{u}nd\ddot{u}zler\ g\ddot{o}rd\ddot{u}m) \approx 6.05$ şeklinde olasılık değeri elde edilir.

2.3.2. Nöral Dil Modelleri

Yüksek boyutlu uzayda noktaların çok az bilgi taşıması, birçok etkisiz değerden (sıfırlardan) oluşmasına vurgu yapan seyreklik (sparsity) kavramı istatistiksel dil modelleme yöntemlerinde yüksek boyut laneti (curse of dimensionality) olarak bilinen probleme yol açmaktadır. Yumuşatma tekniklerinin kullanılması, sınıf tabanlı n-gramlar gibi yöntemler sayesinde nispeten daha iyi sonuçlar elde edilmiştir [42]. Bu problemlerin çözümündeki en önemli gelişme ise vektörlerin seyrek olmasının önüne geçecek, daha anlamlı yoğun (dense) temsillerin elde edilmesidir. Bu amaçla geliştirilen nöral dil modelleme (neural network language modelling) yaklaşımında yapay sinir ağı (YSA) temelli öğrenme teknikleri kullanılmaktadır. Bu teknikler sadece standart dil modelleme ve dil modelleme yöntemlerinin birincil olarak kullanıldığı konuşma tanıma, otomatik tamamlama (autocomplete), sohbet robotu (chatbot) konularında değil ayrıca bir çok DDİ probleminde kullanılmaktadır. Dilin gramer kuralları, kelimeler arasındaki anlamsal ve yapısal ilişkiler gibi çeşitli özellikler otomatik olarak öğrenildiği için temsil öğrenme konusunda da nöral dil modellerinden yararlanılmaktadır. Bu bölümde, nöral dil modelleme tekniklerinin anlatımına yer verirken YSA'lar için bazı temel bilgilerin açıklanması gerekmiştir. Bu bilgiler DDİ işleme yöntemleri perspektifiyle verilmiştir.

2.3.3. Doğal Dil İşlemede Yapay Sinir Ağları

Yapay zekâ, insanlara ait olan düşünme yeteneğinin bilgisayarlara kazandırılmasıyla ilgilenir. Yaklaşık son on yıldır yapay zekânın kullanıldığı hemen hemen her alanındaki problemlerin çözümünde derin öğrenme tabanlı teknikler kullanılmaktadır. Derin öğrenme, yapay zekânın bir kolu olan makine öğrenmesinin bir alt dalıdır ve temelini YSA modelleri oluşturur.

Makine öğrenmesinde, bilgisayar programlarının verilerden otomatik olarak yararlı örüntüler elde etmesine ve geçmiş tecrübelerden yararlanmasına imkân tanıyan yöntemler geliştirilir. Bu yöntemler sayesinde çeşitli görevlerin yapılmasında insanların dahil olmasına minimum düzeyde veya hiç ihtiyaç duymayan sistemler üretilir. Samuel [43] makine öğrenmesiyle ilgili olarak bir bilgisayarın dama oyununu oynayabilecek şekilde programlanabileceğini ve hatta bu oyunu programı yazan programcıdan bile daha iyi oynayabileceğini söylemiştir. Gerçek referans (ground-truth) bilgisinin mevcut olup olmaması ve mevcut olması durumunda nasıl kullanıldığına göre makine öğrenmesi algoritmaları denetimli (supervised), denetimsiz (unsupervised) ve pekiştirmeli (reinforcement) öğrenme olarak üç ana başlık altında gruplandırılabilir [44]. Gerçek referans, modelin belirlenen parametreler için üretmesi istenen sonuçtur. Matematiksel olarak belirli x değerlerine karşı bilinen y değerleridir. Bu bilginin mevcut olması veri setinin etiketli veri seti olduğunu gösterir. Etiketli veri seti üzerinde gerçekleştirilen öğrenmeye ise denetimli veya gözetimli öğrenme denir. Tablo 2.11'de x, y değerlerinden oluşan etiketli bazı örnek veri setleri ve denetimli öğrenme senaryoları verilmiştir.

Tablo 2.11. Etiketli veri kullanılan örnek problemler

Problem türü	**Giriş verisi (x)**	**Etiket (y)**
Metin sınıflandırma	Haber metni	Haberin hangi kategoride olduğu (ekonomi, spor, siyaset vb.) bilgisi
Varlık ismi tanıma	Yer, kişi, kurum adı içeren çeşitli cümleler	Kelimelerin hangi tür varlık ismi olduğunun elle işaretlenmesi
Sözlük türü işaretleme	Çeşitli cümleler	Cümlelerde yer alan isim, fiil, sıfatların elle işaretlenmesi
Makine çevirisi	A dilinde yazılmış doküman	Aynı dokümanın B dilindeki tercümesi
Metin özetleme	Dokümanlar	Dokümanların insanlar tarafından yazılmış özetleri
Duygu analizi	Film izleyicilerinin yorumlarından oluşan cümleler.	Her bir cümlenin duygu bakımından negatiften pozitife doğru ölçekli 1 ile 10 arasında puanlanması

Etiketli verinin kullanılmadığı öğrenme biçimine denetimsiz veya gözetimsiz öğrenme denir. Denetimsiz öğrenme çeşitli nedenlerle tercih edilebilir. Örneğin kullanıcı yorumlarının hangi konulardan oluştuğunun tespit edilmesi probleminde veri olarak sadece yorum metinleri mevcutsa denetimsiz bir öğrenme yöntemi seçilir. Bunun için bir konu modelleme algoritması olan Gizli Dirichlet Ayırımı (GDA) veya kümeleme algoritması olan K-Means kullanılabilir.

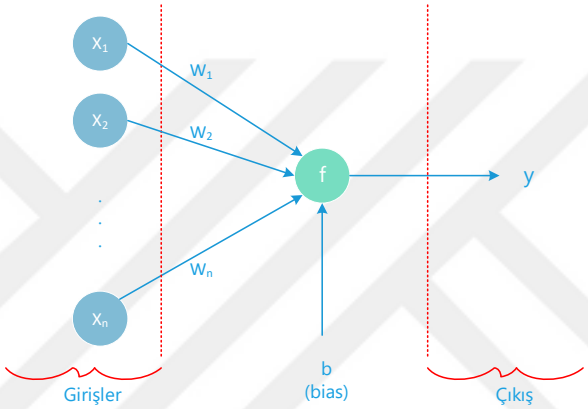
Tablo 2.12. YSA Tabanlı ilk DDİ çalışmaları

Yıl	Çalışmanın konusu
1989	Sonlu durum gramerlerinin basit özyineli yapay sinir ağlarıyla öğrenilmesi çalışması. Bu çalışmayla dilin gramer yapısının YSA'larla öğrenilebilir olduğu gösterilmiştir [45].
1990	İsim, fiil gibi sözcüksel kategorilerinin ve kelimeler arasında çeşitli semantik hiyerarşilerin otomatik olarak tespit edilmesi [46]
1991	Yinelemeli YSA ile dağıtık temsillerin elde edilmesi [47]
1991	Yer edatlarını ve birlikte kullanıldığı kelimelerin semantik özelliklerini, İngilizce ve Almanca için ayrı ayrı eğitilen YSA üzerinden öğrenerek bu modeller arasında semantik temsiller üzerinden otomatik çeviri gerçekleştirme [48]
1993	Sözcük türü tümevarımı (part-of-speech induction) [49]
1994	Cümlelerin İngilizce'den Sırp-Hırvatça'ya çevrilmesi [50]
1996	Dağıtık YSA modeli ile cümle çözümleme, durum-görev(case-role) tespiti [51]

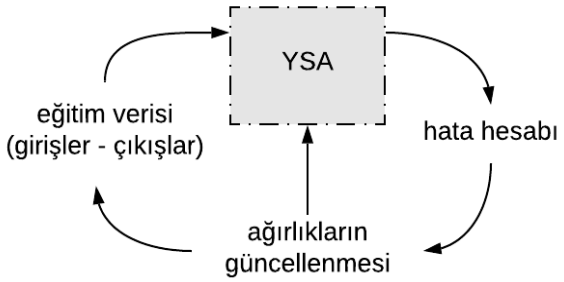
Verilere ait gerçek referans bilgilerinin olmaması veya bu bilgilerin elde edilmesinin çok fazla insan eforu gerektirmesi gibi zorluklar dışında da denetimsiz öğrenme tercih edilebilir. Denetimsiz öneğitme (pretraining) ile modelin belirli bir çıktıyı verecek şekilde şartlandırılmamasının bir avantajı olarak verilerle ilgili yararlı olabilecek birçok farklı özelliklerin öğrenilmesi mümkündür. Denetimsiz öğrenmede bilinen anlamda bir etiket olmamasına rağmen öğrenme sürecinde verinin içerisinde yer alan bazı bilgiler etiket olarak kullanılıyorsa buna öz-denetimli (self-supervised) öğrenme denir. Örneğin sonraki kelimenin tahmin edilmesi probleminde cümlede yer alan kelimelerin sırası bilinir fakat bu bilgi modelden gizlenerek tahmin ettirilir ve model kendisini bu tahminin doğruluğuna göre günceller. Verilerin sadece küçük bir bölümünün etiketli olması durumunda ise iki yaklaşımdan da yararlanılabilir. Bu öğrenme metoduna yarı denetimli (semi-supervised) öğrenme denir. Pekiştirmeli öğrenmede ise bir ödül mekanizması yürütülür. Markov karar süreçleri şeklinde formüle edilen model, doğrudan etiket değerine adapte olmak yerine genellikle sayısal bir değer olan ödülü maksimum seviyede elde etmeye çalışır. Hangi durumda, hangi karar verildiğinde modelin nasıl bir ödül elde edeceği hesaplanır. Bunun için sadece mevcut iterasyondaki (x^i, y^i) bilgisine göre değil, bütün $\{(x^1, y^1), (x^2, y^2) \dots (x^N, y^N)\}$ değerleri için elde edilen ödül değerlerini hesaba katarak bir strateji geliştirilir. Örnek olarak: soru-cevap ve diyalog sistemleri gibi problemlerde sadece

mevcut girdinin değil, veri setindeki tüm girdilerin karşılıklarının dikkate alınması bir pekiştirmeli öğrenme yaklaşımıdır.

Doğal dil işleme problemlerinde seksenli yılların sonlarıyla birlikte YSA yöntemlerinin kullanılmaya başladığı görülmektedir [52]. Bu çalışmaların ilk örneklerinden birisi düzenli ve düzensiz İngilizce fiillerin geçmiş zaman hallerinin YSA ile tahmin edilmesi çalışmasıdır [53]. Geliştirilen modelin bu gramer yapısını öğrenme konusundaki yeteneği küçük çocukların aynı konuda göstermiş olduğu gelişim süreciyle kıyaslanmış ve birbirine paralel özellikler gösterdiği kanısına varılmıştır. YSA'ların DDİ alanındaki problemlerin çözümünde kullanılmaya başlanması açısından önem teşkil eden önemli bazı diğer çalışmalar Tablo 2.12'de verilmiştir.



Şekil 2.6. Nöron örneği



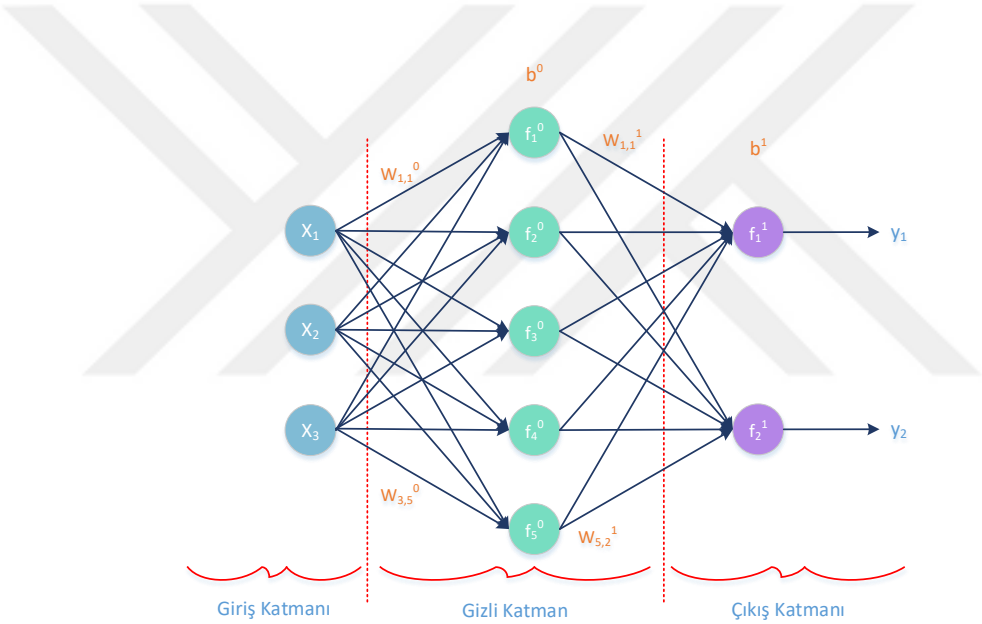
Şekil 2.7. Bir nöronun öğrenme aşamaları

YSA'lar insan zekâsından esinlenerek beynin biyolojik yapısındaki nöronların davranışını taklit etme fikrine dayanır. İnsan beyni nöron adı verilen milyarlarca sinir hücresinden oluşur. Benzer şekilde YSA'lar da nöron adı verilen matematiksel olarak tanımlanmış birimlerden oluşur. Bu birimler katmanlar oluşturacak şekilde bir araya getirilerek çeşitli yapılarda sinir ağları tasarlanır. Giriş ve çıkış katmanları arasında çeşitli hesaplamalar gerçekleştirilerek iteratif bir öğrenme (model eğitimi) sağlanır. Bu yöntemlerin temelini perseptron (algılayıcı) öğrenme

algoritması oluşturur. Şekil 2.6'da $x_1, x_2, \ldots x_n$ değişkenlerinden oluşan giriş katmanı ve bu değişkenler için sırasıyla $w_1, w_2, \ldots w_n$ ağırlık (weight) değerleri tanımlanan örnek bir nöron gösterilmiştir.

Giriş değerleri bir bias (b) değeriyle toplanarak f fonksiyonundan elde edilen değer çıkışa iletilir. Algılayıcı (perceptron) olarak da isimlendirilen nöron YSA'ların en temel birimidir. Şekil 2.6'da gösterildiği üzere x girişleri ve önceden bilinen çıkış $\hat{\mathbf{y}}$ için w parametreleri ayarlanırken her adımda bir $\mathbf{y} = f(x_1 w_1 + x_2 w_2 + \cdots + x_n w_n + b)$ değeri hesaplanır. Bu öğrenme çevrimi Şekil 2.7'de gösterilmiştir.

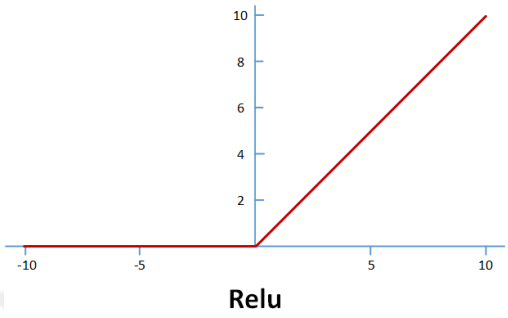
Nöronlar bir araya getirilerek sinir ağı elde edilir. Nöronların hepsinin aynı yönde iletime geçtiği, çevrim içermeyen ağlara İleri Beslemeli Sinir Ağları (Feedforward Neural Network - FFNN) denir. Bir FFNN Şekil 2.8'de gösterildiği gibi giriş, gizli ve çıkış katmanlarından oluşur.



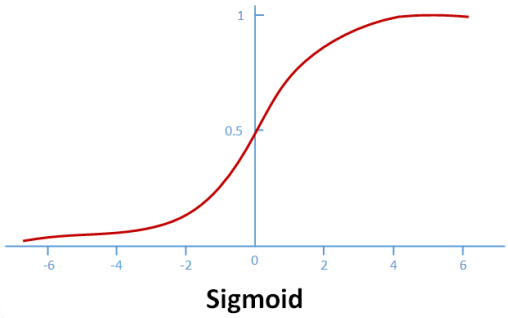
Şekil 2.8. İleri beslemeli sinir ağı yapısı

Bir nöronun kendisine giriş olarak gelen değerlerin ağırlıklı toplamlarını devam eden nöronlara aktardığı aşamada $\boldsymbol{f}$ ile belirtilen **aktivasyon fonksiyonu** devreye girer. Biyolojik nöronlarda, gelen bir sinyal bazı nöronların birbiri ardına iletime geçmesiyle beyne iletilirken bazı nöronlar hiç devreye girmez. Bu şekilde koku, tat gibi farklı duyu alıcıları ilgili olduğu beyin bölgesini başarıyla uyarır. YSA'lar da benzer şekilde bir aktivasyon mekanizması (ReLu, TanH, SoftMax, sigmoid, binary step vb.) işletilir. Nöron aktivasyonları bu iletimleri ölçekleyebilir veya aktif/pasif davranışı gösterebilir. Örneğin nöronların lineer bir şekilde gelen değer büyüdükçe daha çok aktive olmalarını sağlarken sıfırdan küçük değerlerde pasif olmalarını sağlayan f : $\max(x, 0)$ fonksiyonu kullanılır. Bu fonksiyon YSA'larda ReLu (Rectified Linear Unit) olarak bilinmektedir. Literatürde bilinirliği yüksek 10'un üzerinde farklı aktivasyon fonksiyonu çeşidi

bulunmaktadır. Bunlardan bir diğeri de $f : 1 / (1 + e^{-x})$ sigmoid fonksiyonudur. Sigmoid, modern YSA tabanlı yöntemlerde oldukça sık kullanılan, giriş değerleri ne olursa olsun 0 ile 1 arasında çıkışlar üreten lineer olmayan bir fonksiyondur. Şekil 2.9 ve Şekil 2.10'da bu fonksiyonların davranışı çizdirilmiştir.



Şekil 2.9. Relu fonksiyonu



Şekil 2.10. Sigmoid fonksiyonu

Mevcut iterasyondaki parametrelerle elde edilen hata değeri $\hat{\boldsymbol{y}} - \boldsymbol{y}$'dir. Tepe tırmanma algoritmalarının bir çeşidi olan **gradyan iniş** (gradient descent) optimizasyon algoritması ile bu hata minimize edilerek bir öğrenme gerçekleştirilir. Her adımda karesel hata gibi bir doğruluk ölçütü üzerinden yinelemeli olarak parametre optimizasyonu gerçekleştirir. Elde edilen hatanın $(\boldsymbol{J})$ türevi alınarak hatanın yönü ve miktarı belirlenir. Bu değer önceden belirlenen bir öğrenme katsayısı (α) ile çarpıldıktan sonra parametrelere eklenir. Bu şekilde optimizasyon tamamlanana kadar her iterasyonda $\boldsymbol{w_i}$ parametreleri güncellenir. Öğrenme katsayısı, hata fonksiyonu, aktivasyon fonksiyonu gibi tercihlerin nasıl yapılacağının belirli bir kuralı yoktur. Bu parametreler her problem için ayrı ayrı belirlenir. Bahsedilen adımlar ve zincirleme türev alma işlemi Denklem 2.7'de gösterildiği şekilde gerçekleştirilir.

$$s = \sum_{i=1}^{n} x_i w_i + b \tag{2.7}$$

$$y = ReLu\,(s)$$

$$J = \frac{1}{2}\,(\hat{y} - y)^2$$

$$\frac{dJ}{dw_i} = \frac{dJ}{dy} \cdot \frac{dy}{ds} \cdot \frac{ds}{dw_i}$$

$$w_i \leftarrow w_i - \alpha \frac{dJ}{dw_i}$$

2.3.4. Nöral Dil Modeli ve Dağıtık Temsiller

Bir kelimenin anlamı sözlükte o dile ait başka kelimeler üzerinden açıklanır ve yine başka kelimelerle birlikte kullanıldığı örnek cümlelerle zihnimizde bir yer edinir. Örneğin *spagetti* kelimesi "*Çeşitli soslarla yapılan İtalyan makarnası*" olarak tanımlanır. Cümle içinde ise "*akşam yemeği menüsünde spagetti ve salata var*" şeklinde kullanılır. Makinaların benzer bir öğrenme sürecini taklit etmeye en çok yaklaştığı yöntemler YSA tabanlı modellerle elde edilen dağıtık temsillerdir. Boyutu önceden tanımlı bir uzayda *spagetti* kelimesinin yemek anlamı dolayısıyla *makarna, salata, sos* gibi kelimelerle birbirine yakın olması, menşe adı bakımından *Gürcistan* kelimesine *İtalya* kelimesinden daha uzak olması oldukça akla yatan bir sonuç olacaktır.

YSA'ların dil modellemede kullanılması ile standart n-gram dil modellerine ait boyutluluk laneti gibi problemlerin önüne geçilebileceği öngörülmüş ve aynı zamanda getirdiği yeni yaklaşımlar sayesinde daha gelişmiş dil modellerinin elde edilmesinin önü açılmıştır. FFNN, RNN, LSTM gibi YSA varyantlarının ortaya çıkmasına paralel olarak hesaplama gücündeki gelişmeler ve eğitim veri setlerinin elde edilme olanaklarındaki artış sayesinde çok daha başarılı dil modellerinin geliştirildiğini görmekteyiz. Nöral modeller, konuşma tanıma ve makine çevirisi gibi alanlarda kural tabanlı ve istatistiksel yöntemlerin neredeyse tamamen terk edilmesine yol açmıştır.

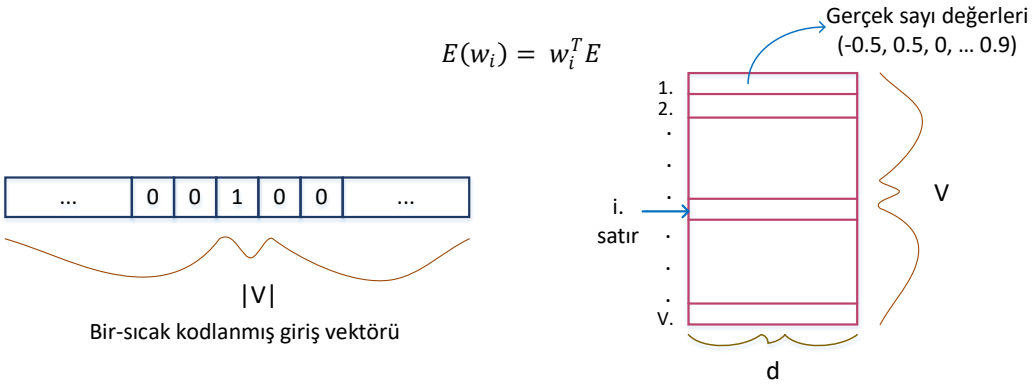
Bu alandaki ilk çalışmalardan birisi Xu ve Rudnicky [54] tarafından tek katmanlı bir YSA ile geliştirilen bigram dil modelidir. Hata fonksiyonu olarak çapraşıklık değerinin logaritması kullanılmıştır. Kullanılan sinir ağının tek katmanlı olması modelin genelleme açısından zayıf olmasına neden olsa da yumuşatma tekniklerinin uygulandığı bazı istatistiksel yöntemlere göre daha iyi sonuçlar elde edilmiştir.

Bengio ve arkadaşlarının önerdiği, sırasıyla giriş, haritalama (mapping), gizli ve çıkış katmanlarından oluşan ileri beslemeli sinir ağı ile bir yandan dil modeli öğrenilirken bir yandan da sürekli bir uzayda dağıtık kelime temsilleri öğrenilmiştir [24]. Literatürde klasik nöral dil modeli olarak bilinen bu yöntemle 15 milyon kelimeden oluşan bir derlem üzerinde yumuşatma teknikleriyle uygulanan geleneksel yöntemlere göre çapraşıklık değeri bakımından %10-%20 arasında daha iyi sonuçlar veren bir dil modeli elde edilmiştir.

Nöral dil modellemede geleneksel n-gram dil modellerinde olduğu gibi her bir kelime (w_i) kendisinden önce gelen $w_{i-(n-1)}$ ile w_{i-1} aralığındaki n-1 tane kelime baz alınarak tahmin edilir. Tahmin hesabı, n-gram dil modelinden farklı olarak temel sayma prensibi üzerinden değil, bir YSA ile yapılır. Ön koşul olarak kullanılan kelimelere bağlam (context) denir. Bu tez çalışması boyunca bağlam olarak kullanılan ön koşul kelimelerini belirtmek için *pencere* kelimesi tercih edilmiştir. Klasik nöral dil modelinde bir pencerenin sağındaki kelime tahmin edilir fakat Bölüm 2.4'te verilen yöntemlerde görüleceği üzere pencereyi oluşturan kelimeler farklı şekillerde seçilebilir. Seçilen her pencere için sırada gelen kelimenin olasılığı maksimize edilir. Denklem

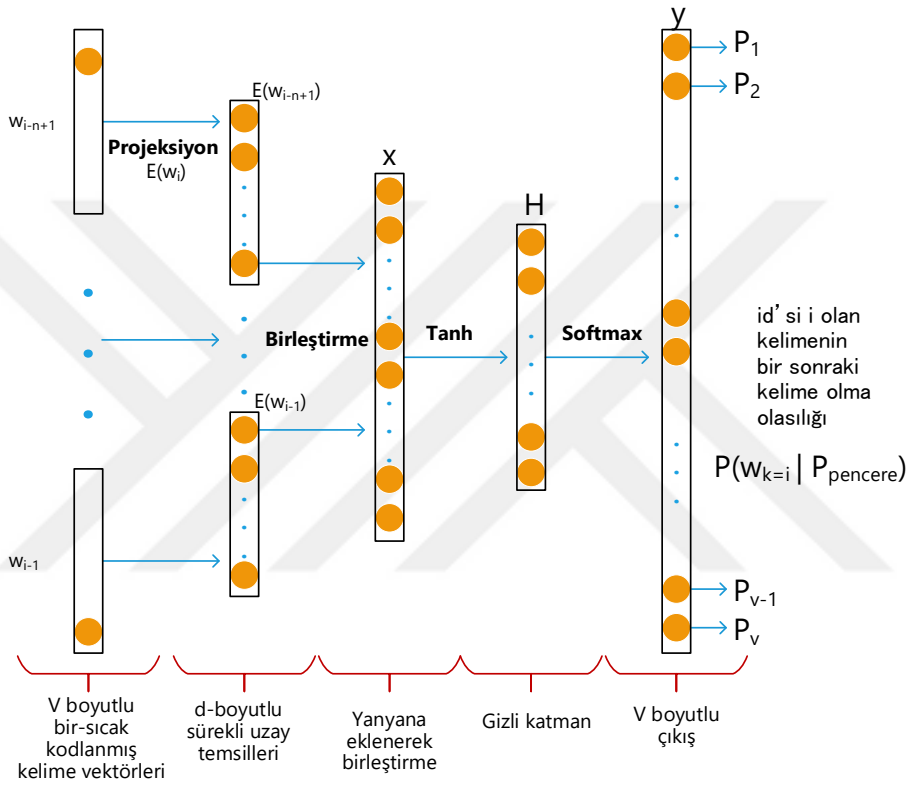
2.4'e uyarlanırsa belirtilen olasılık $P(w_i \mid w_{i-(n-1)}, \ldots w_{i-1})$ yerine $\boldsymbol{P(w_i \mid pencere)}$ olarak da gösterilebilir. Giriş katmanından sinir ağına verilen pencereden sonra hangi kelimenin hangi olasılıkla geleceği çıkış katmanında sözlükteki bütün kelimeler için hesaplanır. Standart FFNN öğrenme adımları olan stokastik gradyan iniş (SGD) [55] ve hatanın geri yayılımıyla her iterasyonda ağırlıklar güncellenir. Sinir ağına giriş olarak verilen bütün pencereler için doğru n-gram koşullarını sağlayan maksimum olabilirlik dağılımları elde edildiğinde eğitim tamamlanır. Model, asıl amacı olan eğitim verisi içerisindeki pencerelerin sonraki kelimelerini tahmin etmeyi öğrenmeyi tamamladığında sözlükte yer alan her bir kelime için $\boldsymbol{d}$ boyutlu bir vektör elde etmiş olur. Modelin başarısı test verisi üzerinde çapraşıklık gibi bir metrik üzerinden gösterilebilir, fakat genellikle bu modelin başka bir görev için kullandığında nasıl başarı gösterdiğine bakılarak bir değerlendirme tercih edilmektedir. Örneğin makine çevirisinde gösterdiği performansa göre değerlendirilebilir [56]. Modele ait diğer önemli detaylar şu şekildedir:

Boyutu $\boldsymbol{V}$ olan sözlükteki her kelimeye 1,2,3, … V'ye kadar bir numara (id) atanır. Kelimeler giriş katmanına V boyutunda bir-sıcak kodlu vektörler $\forall w_i \in [1, V]$ olarak verilir. Eğitim sırasında bu katmandaki bir-sıcak temsiller değişmez. Bir sonraki katmanda, her bir kelimeye karşılık bir vektör tutan $|V|xd$ boyutlu bir izdüşüm matrisindeki değerler öğrenilir. Şekil 2.11'de bu matris $\boldsymbol{E}$ ile, içerdiği her bir kelimenin vektör karşılığı ise $E(w_i) \in \mathbb{R}^d$ ile gösterilmiştir. Öğrenilen kelime vektörlerine **gömme** (embedding) denir. Gömmeler, boyutu önceden belirlenen yoğun vektörlerdir, fakat diğer kelime temsil vektörlerinden farklı olarak bir makine öğrenmesi yöntemiyle öğrenilmiş olma özelliği taşırlar. Kaliteli kelime gömmelerinin öğrenilmesi DDİ alanındaki birçok konu için önemli bir ilgi alanı haline gelmiştir. Kelimelerin önce bir-sıcak kodlanıp daha sonra E matrisindeki değerlerinin bulunmasıyla otomatik bir arama (look-up) tablosu yapısı elde edilir.



Şekil 2.11. Kelime vektörü gösterim şekilleri

Şekil 2.12'de id'si i olan bir kelimenin $|V|xd$ boyutlu matristeki gerçel sayılardan oluşan vektör değerinin nasıl bulunduğu gösterilmiştir. Bağlam içerisindeki $n-1$ adet kelime yan yana getirilerek gizli katmanın girişi olan x vektörü elde edilir. Gizli katman (H) bu vektörün $\tanh(x)$ (hiperbolik tanjant) değerini hesaplar ve sonucu çıkış katmanına aktarır. Çıkış katmanında gelen değer softmax aktivasyonundan geçerek bir sonraki kelime için olasılık dağılımı (y) bulunur. Bu aşamalar Denklem 2.8'de gösterilen adımlarla özetlenmiştir.



Şekil 2.12. Nöral dil modelinin eğitim aşamaları

$$x = \left(E(w_{i-1}), E(w_{i-2}), \dots, E(w_{i-(n-1)})\right) \qquad x \in \mathbb{R}^{(n-1)xd} \tag{2.8}$$

$$H = \tanh(x)$$

$$s = \theta \cdot H \qquad (\theta: gizli\ katman\ ağırlıkları)$$

$$y = softmax(s)$$

Modelin son katmanında mevcut parametrelerle bir tahmin yapılması gerekir. Bu aşamada genellikle **softmax fonksiyonu** tercih edilerek bir maksimum olasılık dağılımı elde edilir. Sözlükteki her bir kelime, denetimli öğrenme probleminde yer alan kategorik sınıflar gibi

düşünülür ve toplamları 1 olacak şekilde her sınıf için bir olasılık değeri bulunur. K adet kategori için genel softmax formülü Denklem 2.9'da verilmiştir.

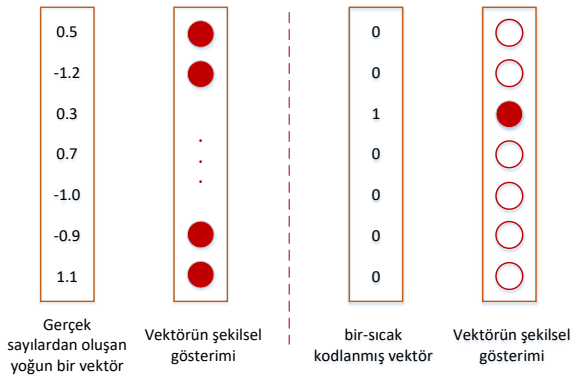
$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{k=1}^{K} e^{z_i}} \tag{2.9}$$

Denklem 2.9'daki K, sözlüğün boyutu olan V büyüklüğündedir. Beş kelimeden oluşan bir sözlük için örnek değerler ve bu değerlerden elde edilen softmax dağılımı Tablo 2.13'de gösterilmiştir.

Tablo 2.13. Çıkış katmanı için örnek softmax aktivasyon hesabı

Kelime (w_i)	**Değer**	**e^{w_i}**	**$\sigma(w)_i$**
$w_{i=1}$	-1.5	0.2231	0.0054
$w_{i=2}$	2.0	7.3891	0.1799
$w_{i=3}$	-1.1	0.3329	0.0081
$w_{i=4}$	-4.0	0.0183	0.0004
$w_{i=5}$	3.5	33.1155	0.8061
TOPLAM			1.0000

Tez içerisindeki vektör gösterimleri Şekil 2.13'teki gibidir.



Şekil 2.13. Bir Vektör gösterim formatı

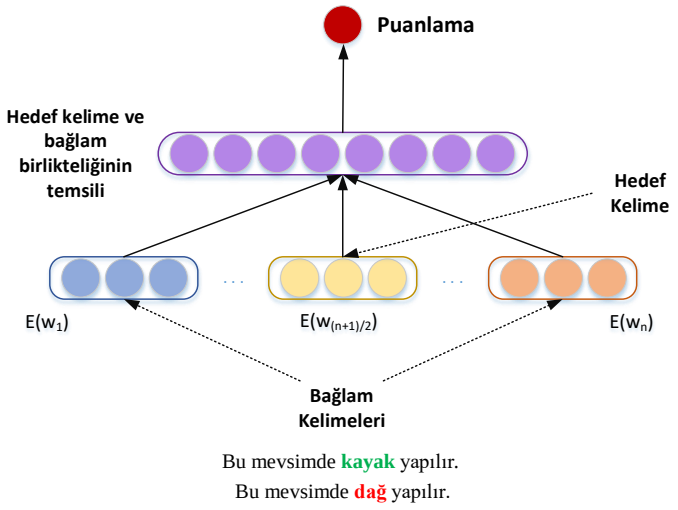
Dağıtık temsillerin kullanımı hem dil modellemede hem de birçok DDİ probleminin çözümünde performans artışlarını tetikleyen önemli bir etki yaratmıştır [57]. Dil modeli öğrenildikten sonra elde edilen kelime temsilleri metin sınıflandırma, duygu analizi gibi farklı görevler için kullanılmaktadır.

Yerelci yaklaşımda öğrenilmek istenen her kavram için ayrı bir sinir ağı düğümü görevlendirildiğinden dolayı sonlu bir vektör uzayında oldukça sınırlı bir öğrenme gerçekleşir [58]. Özellikler ayrıktır ve birbirinden bağımsız olarak ele alınır. Yani düğümler sadece kendisine adanan kavramın temsilinde aktive olur. Tasarım ve uygulama yönünden oldukça basit olan bu yöntem zengin ve karmaşık bir yapı olan dile ait kelimelerin farklı sözdizimsel ve semantik bileşenlerini öğrenmek konusunda yetersiz kalmaktadır. Dağılımsal yaklaşımda ise temsil edilen kavramlar ve düğümler arasında çoka-çok (many to many) bir ilişki kurulur ve öğrenilen özellikler düğümler arasında dağılır [47].

2.4. Word2Vec

Bengio ve arkadaşları [24] tarafından önerilen FFNN ile nöral dil modeli eğitimine paralel olarak bir yandan dağıtık kelime temsilleri öğrenilmektedir. Tahmine dayalı kelime temsillerinin elde edilmesi yaklaşımındaki en önemli gelişmelerden biri olan bu yöntem, öncesinde önerilen diğer dil modellerine göre daha başarılıdır fakat sadece kelime temsili öğrenmek için kullanılması durumunda verimli olmayacaktır. Bu yöntemin başlıca geliştirilmeye açık yönleri ve bunlara karşı literatürde önerilen yenilikler şunlardır:

- Model, sadece bir sonraki kelimeyi tahmin edecek şekilde kurgulanmıştır. Bu anlamda geleneksel istatistiksel n-gram dil modeli yaklaşımına benzemektedir. Bunun yerine baz alınan pencere içerisindeki kelimelerin birbiriyle olan anlamsal ve dilbilimsel bağlarını daha kapsamlı keşfetmek için farklı mimariler önerilmiştir.
- Projeksiyon katmanı ile gizli katman arasında çok fazla parametrenin hesaplanması modelin karmaşıklığını arttırmaktadır. Bu aşamada, paralel hesaplama kullanarak performans iyileştirilmesi ile gizli katmanın kullanılmaması gibi öneriler getirilmiştir.



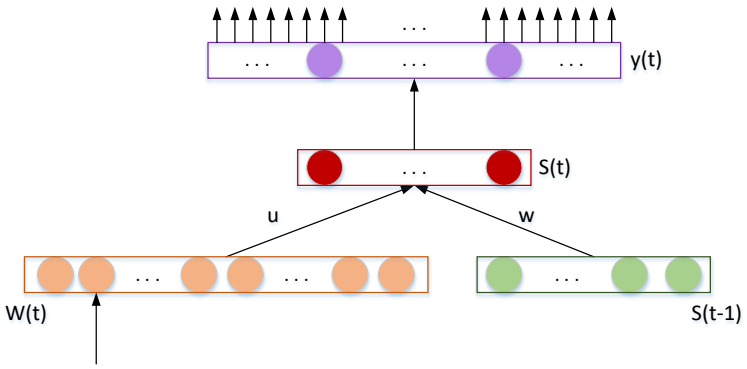
Şekil 2.14. C&W temsil yöntemi [59]

R. Collobert ve C. Weston (C&W) yalnızca kelime gömmesi eğiten bir model önermiştir [60]. Milyonlarca kelime içeren büyük derlemler üzerinden temsil öğrenebilme yeteneğine sahip olan C&W modeli, hedef kelimeyi tahmin etmek üzerine kurgulanmamıştır. Seçilen merkezdeki kelime ve etrafındakilerin (bağlam) bir arada bulunmalarını doğru tahmin etmek üzere bir gizli katman eğitilmektedir. Derlem içerisinde yer alan kelime dizileriyle birlikte, rasgele seçilen, bağlamda yer almayan kelimelerle hatalı örnekler üzerinden bir öğrenme söz konusudur. Şekil 2.14'te C&W modelinin temel yapısı gösterilmiştir.

Mikolov ve arkadaşları [61] Tekrarlayan Sinir Ağı Dil Modelini (Recurrent Neural Network Language Model - RNNLM) önermiştir. Giriş katmanı, bir adet gizli katman, tekrarlı bağlantılar ve ağırlıkların saklandığı matrislerden oluşan bu mimari ile her iterasyonda softmax fonksiyonu ile kelime tahmini yapılır. RNNLM'nin çalışma prensibi Şekil 2.15'te gösterilmiştir. Bir t anındaki (t. iterasyon) kelimenin bir-sıcak kodlanmış giriş vektörü $w(t)$, kısa süreli bir hafıza gibi işlev gören gizli katman ağırlık değerleri $s(t)$ olmak üzere; her kelime için dağılımsal olasılık değeri sonuçları $y(t)$ şu şekilde hesaplanır:

$$s(t) = sigmoid\,([\Theta w(t),\ \Upsilon s(t-1)]) \tag{2.10}$$

$$y(t) = \text{softmax}(Hs(t)) \tag{2.11}$$

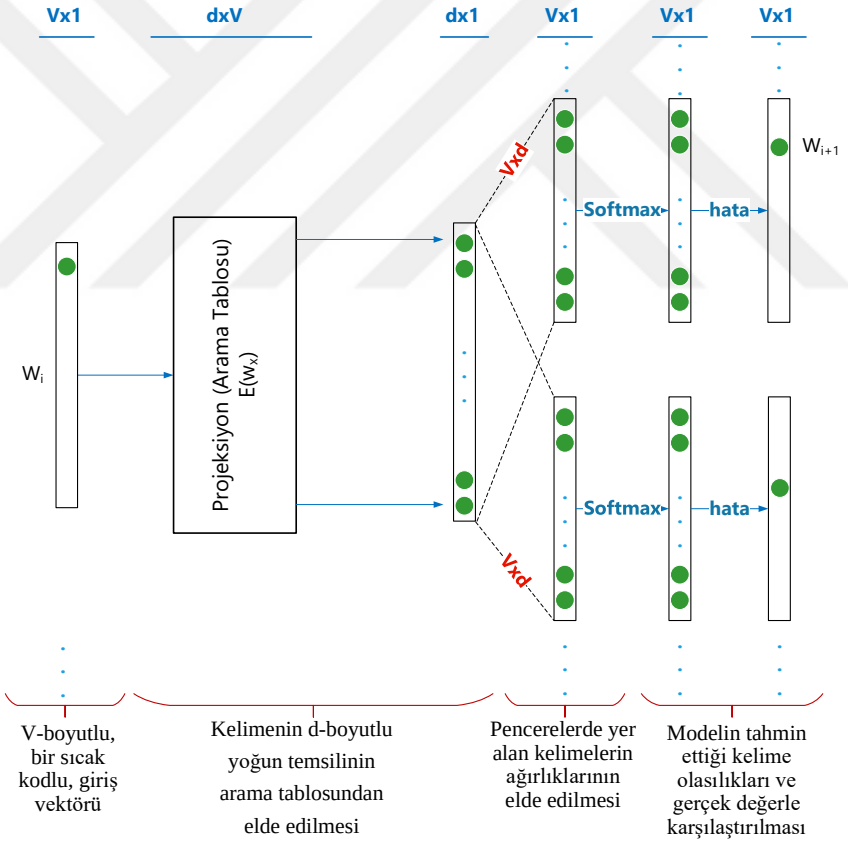


Şekil 2.15. RNNLM

Denklem 2.10'da gösterildiği üzere gizli katman girişine o anki kelime temsili ile kendisinin bir önceki kelime için ürettiği çıkış değerleri birleştirilerek verilmektedir. Görünüşte sadece bir önceki kelimeye bağlı bir olasılık tahmini yürütülüyor olsa da bu işlem o ana kadarki her kelime için tekrarlayan bir şekilde uygulandığı için hafızalı bir işleyiş söz konusudur. Bu da RNNLM'nin FFNNLM'lere göre en önemli avantajından birisidir. Kelimeler arasındaki çeşitli semantik ve sözdizimsel ilişkilerin bu yöntem ile elde edilen kelime vektörleri arasında da mevcut olduğu görülmüştür. Anlamsal benzerlik konusundaki başarının ölçülmesi için SemEval-2012

görevi [62] gibi değerlendirmeler kullanılmıştır. Bu çalışma; "Türkiye" ve "Ankara" kelimeleri için elde edilen vektörler arasındaki ilişkinin "İngilitere" ve "Londra" kelimelerinin vektörleri için sağlanması gibi analojilerin popülerleşmesine neden olmuştur. Bu analojiler "kalem" – "kalemler" ile "kitap" – "x" için "kitaplar" cevabını veren farklı sözdizimsel özellikleri de kapsamaktadır.

FFNNLM ve RNNLM'den daha basit bir mimariye sahip, gizli katmanında lineer olmayan aktivasyon işlemi ihtiva etmeyen Skip-Gram ve sürekli kelime-torbası (continuous bag-of-words – CBOW) algoritmaları ile büyük veri setlerinden daha kaliteli kelime vektörlerinin daha hızlı bir şekilde elde edilmesi mümkündür [63]. Literatürde, bu iki algoritmayı kapsayan çatı bir terim olarak implemantasyonları için geliştirilmiş açık kaynaklı yazılım aracının adı olan word2vec [64] de kullanılabilmektedir.



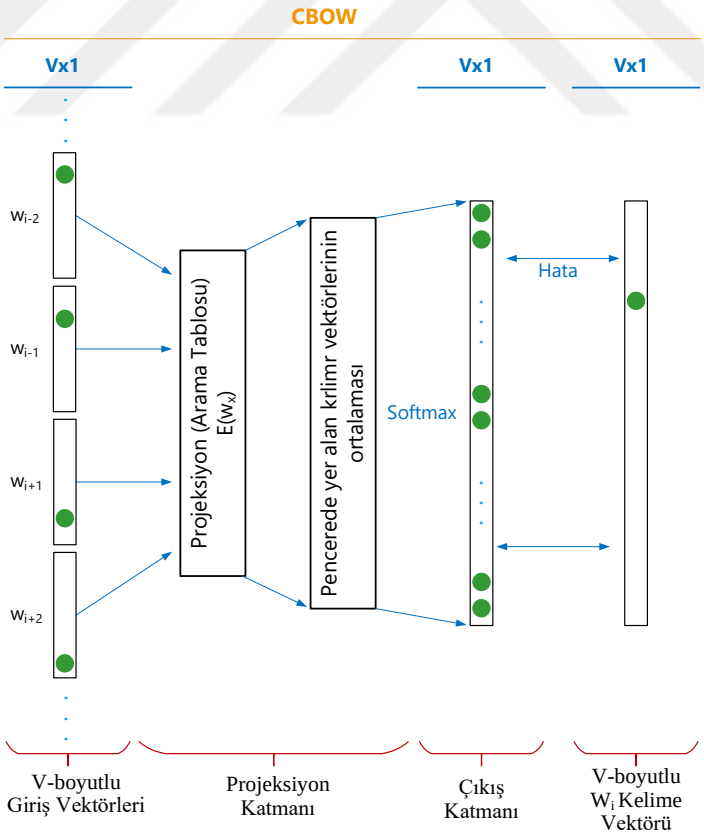
Şekil 2.16. Skip Gram

Skip-gram ve CBOW yöntemlerinde son katmanda tahmin edilen kelime veya kelimelerin doğruluğu hesaplanırken hata değeri olarak negatif log olabilirliği (negatif log-likelihood) kullanılır. Ayrıca bu iki algoritma orijinal makalelerinde log-lineer modeller olarak

anılmaktadır. Eğitimi sırasında güncellenen parametreleri θ_i, giriş verisine ait x özelliklerinin fonksiyonu $f_i(x)$ olmak üzere, bir log-lineer model şu şekilde gösterilebilir [65]:

$$C\ e^{\sum_i \theta_i f_i(x)} \tag{2.12}$$

Denklem 2.12'de verilen modelin logaritması alındığında model parametrelerinin lineer (doğrusal) bir fonksiyonu elde edilir ve bundan dolayı böyle modeller log-lineer olarak adlandırılır. Denklem 2.11'de verilen softmax fonksiyonu da özel bir log-lineer modelidir. Skip-gram algoritmasında bir kelime (w_i) giriş olarak verilip çıkış katmanında aynı pencerede bulunan diğer kelimelerin (w_{i-n}, …, w_{i-2}, w_{i-1}, w_{i+1}, w_{i+2}, … , w_{i+n}) tahmin edilme olasılığını arttıracak şekilde bir eğitim gerçekleştirilir. CBOW algoritmasında ise pencere içerisindeki kelimelerden bir tanesinin çıkış katmanında tahmin edilmesi için aynı pencere içerisindeki diğer kelimeler giriş katmanına verilir. Gizli katmana iletilen kelimelerin işaret ettiği ağırlık değerleri toplanır ve ortalaması alınarak çıkışa verilir. Elde edilen çıkış değeri tahmin edilmekte olan kelimenin bir-sıcak kodlanmış vektörüyle karşılaştırılarak hata değeri elde edilir. Şekil 2.16 ve Şekil 2.17'de word2vec algoritmalarının mimarileri gösterilmiştir.



Şekil 2.17. CBOW

Skip gram modelinde amaç, pencere içerisindeki kelimelerin olasılığını maksimize etmektir. Bunun için hata fonksiyonu olan negatif log olasılıklarının ortalaması minimize edilir. Her bir kelime vektörüne ait parametreler θ, pencere boyutu n olmak üzere derlem içerisindeki N tane kelimeden oluşan bir alt dizideki (eğitim verisi) her kelime (w_i) için aynı pencerede yer alan diğer kelimelerin olasılığı ($\mathcal{L}$) ve buna bağlı amaç fonksiyonu (J) hesaplanır.

$$\mathcal{L}(\theta) = \prod_{i=1}^{N} \prod_{\substack{-n \le j \le n \\ j \neq 0}} P\left(w_{i+j} \middle| w_i; \theta\right) \tag{2.13}$$

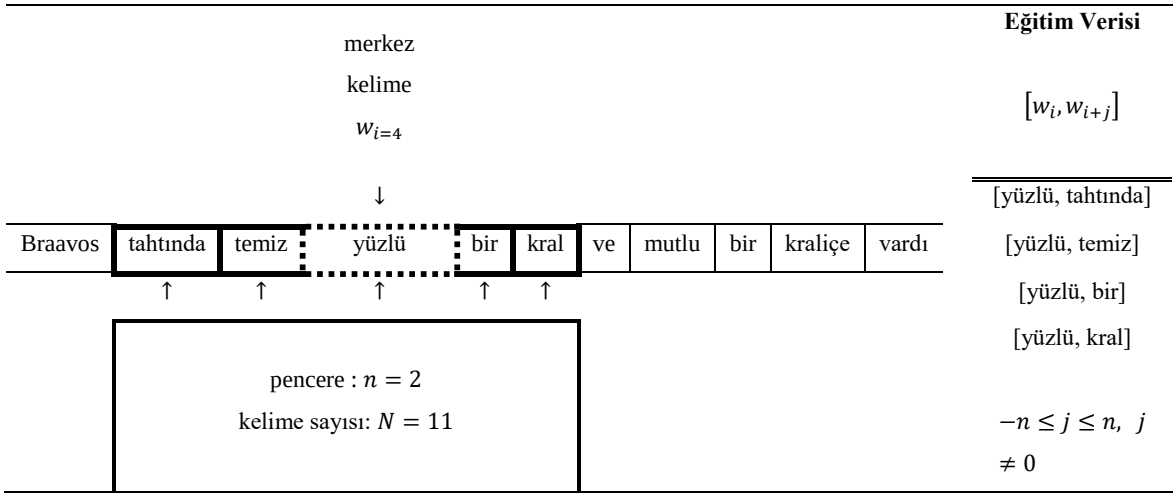
$$J(\theta) = -\frac{1}{N} \log \mathcal{L}(\theta) \tag{2.14}$$

$$= -\frac{1}{N} \sum_{i=1}^{N} \sum_{\substack{-n \le j \le n \\ j \neq 0}} \log P\left(w_{i+j} \middle| w_i; \theta\right)$$

Şekil 2.18'de örnek "Braavos tahtında temiz yüzlü bir kral ve mutlu bir kraliçe vardı" cümlesi üzerinde pencere boyutu 2 seçilmesi durumunda 4. kelime merkez alınarak elde edilen eğitim verileri gösterilmiştir. Kelimelerin giriş katmanı ile gizli katman arasındaki vektör temsilleri v_{w_i}, gizli katman ile çıkış katmanı arasındaki temsilleri v'_{w_i} şeklinde gösterilmek üzere, çıkış katmanında $P\left(w_{i+j} \middle| w_i\right)$ olasılığı softmax ile hesaplanır.

$$P\left(w_{i+j} \middle| w_i\right) = P(w_{i-n}, \dots, w_{i-1}, w_{i+1}, \dots, w_{i+n} \,|w_i) \tag{2.15}$$

$$= \frac{exp({v'_{w_{i+j}}}^{T} v_{w_i})}{\sum_{k=1}^{V} exp({v'_{w_k}}^{T} v_{w_i})}$$



Şekil 2.18. Skip Gram pencere adım

Softmax hesabının bu şekilde sözlükte bulunan her kelime için hesaplanması oldukça maliyetli bir yöntemdir. Mikolov ve arkadaşları bu naif hesaplamayı daha pratik bir şekilde yapmak üzere negatif örnekleme ve hiyerarşik softmax yöntemlerini önermiştir [66]. Hesaplama maliyetini azaltmak için ayrıca metin ön işleme aşamasında frekansı çok yüksek olan kelimeler daha az örneklenerek eğitim verisi sayısı azaltılır. Bu yönteme alt-örnekleme (sub-sampling) denir.

CBOW modelinde giriş ve çıkış katmanları skip-gram modeline göre yer değiştirmiştir. Dolayısıyla amaç fonksiyonunda (J) merkez kelimenin tahmin edilmesinde penceredeki kelimelerin vektör temsillerinin ne kadar doğru sonuç verdiği hesaplanır.

$$J(\theta) = -\frac{1}{N}\sum_{i=1}^{N} log\ P(w_i \mid w_{i-n}, \dots, w_{i-1}, w_{i+1}, \dots, w_{i+n}) \quad (2.16)$$

2.5. Glove

Word2vec algoritmaları, kelimelerin birlikte bulunma koşullarını seçilen bir pencerenin derlem içerisinde gezdirilmesiyle öğrenirken kelimelerin tüm derlem içerisinde ne sıklıkla bir arada bulunduklarını dikkate almaz. Yani birbirinden bağımsız yerel istatistiklerden yararlanır fakat derlemin tamamını kapsayan genel (global) istatistikten yararlanmaz. **GloVe** (Global Vectors for Word Representation) [67] log-bilineer kelime temsili öğrenme modelinde ise yerel birlikteliklerden oluşan bağlam bilgisi yerine matris ayrıştırma yönteminde olduğu gibi global birliktelik frekansları kullanılır. Yani GloVe, Word2vec gibi öngörücü (predictive) bir yaklaşım yerine sayım tabanlı bir temsil öğrenme sunmaktadır.

Kelime birliktelik matrisindeki değerler kullanılarak kelimeler (w_i ve w_j) arasında birliktelik olasılığı $P(w_i|w_j)$ hesaplanır. Bu olasılık word2vec yöntemindeki aynı pencerede bulunma koşulu gibi öğrenilen yoğun vektör temsiline ait değerlerin şekillenmesini sağlar.

Kelime birliktelik matrisindeki değerler kullanılarak kelimeler arasında birliktelik olasılığı hesaplanır. Bu olasılık word2vec yöntemindeki aynı pencerede bulunma koşulu gibi öğrenilen yoğun vektör temsiline ait değerlerin şekillenmesini sağlar. w_i kelimesi bağlamında w_j'nin bulunma sayısı X_{ij} ve derlem içerisindeki bütün kelimeler için w_i bağlamında bulunma sayıları toplamı $\sum_i X_{ik} = X_i$ olmak üzere w_i bağlamında w_j'nin bulunma olasılığı P_{ij} şu şekilde hesaplanır:

$$P_{ij} = P(w_j|w_i) = \frac{X_{ij}}{X_i} \quad (2.17)$$

Bu iki kelimenin üçüncü bir kelime (w_k) bağlamında temsilleri için şu şekilde bir yaklaşım önerilmiştir:

$$F((w_i - w_j)^T \tilde{w}_k) \approx \frac{P_{ik}}{P_{jk}} \quad (2.18)$$

F fonksiyonu, kelime vektörlerine uygulanan fark işleminin transpozunun bağlam kelimesi vektörüyle iç çarpımını alır ve bu değerle bağlam kelimesi için ayrı ayrı hesaplanan birliktelik olasılıkları oranına uymak üzere bir öğrenme gerçekleştirir. Dikkat edilirse, w_k kelimesi semantik benzerlik bakımından w_i ile alakalı olup w_j ile alakasız olduğunda P_{ik}/P_{jk} oranı büyüyecektir. Yani w_i ile w_j kelimelerinin vektörleri arasındaki farkın büyüklüğü sayı tabanlı değerler üzerinden kurulan benzerlik ilişkisine bağlıdır. Bu işlemdeki olasılık değerleri yerine yazılıp bias terimleri dahil edildiğinde şu şekilde basitleştirilebilir:

$$w_i^T \tilde{w}_k + b_i + \tilde{b}_k = \log(X_{ik}) \quad (2.19)$$

Sonuç olarak derlemdeki kelime ikilileri için $w_i,\ w_j \in \mathbb{R}^D$ kelime vektörlerini öğrenen en küçük kareler regresyon modeli için hata fonksiyonu şekilde tanımlanır:

$$J = \sum_{i,j=1}^{V} f(X_{ij})(w_i^T \tilde{w}_j + b_i + \tilde{b}_j - logX_{ij})^2 \quad (2.20)$$

$$f(\mathrm{x}) = min((x/x_{max})^{\alpha}, 1) \quad (2.21)$$

X_{ij} birliktelik sayılarının hata değerine olduğu gibi dahil edilmesi yerine f fonksiyonu ile bir ağırlıklandırmaya tabi tutulur. Böylelikle, bir kelimenin derlemde çok fazla veya çok az bulunması durumlarında modelin gereğinden fazla etkilemesinin önüne geçilir. Örneğin frekansı yüksek olan önemsiz kelimelerin hata fonksiyonunu çok fazla yönlendirmesi sınırlandırılır. Glove modelinin orijinal makalesinde hiperparametlerin $x_{max} = 100$ ve $\alpha = 0.75$ olarak seçilmesi durumunda iyi bir performans sağlandığı belirtilmektedir.

2.6. FastText

Gelişmiş bir word2vec varyantı olan FastText, basit fakat etkili bir kelime gömme yöntemi sunmaktadır [68]. Facebook tarafından geliştirilmiştir ve Word2vec yönteminin başlıca yazarı olan Mikolov [63] FastText çalışmasının da yazarlarındandır. Glove ve Word2vec yöntemlerinde ele alınan en atomik birimler kelimeler olduğu için eğer bir kelime modelin eğitildiği derlem içerisinde yer almıyorsa, gömme değerinin hesaplanması mümkün değildir. Derlemde yer almayan bütün kelimelerin karşılığında ortak olarak kullanılmak üzere tek bir vektör değeri hesaplanmaktadır.

"Buzdolabında", "elma", "portakal", "var" kelimelerinin bulunduğu fakat "papaya", "kinoa" kelimelerinin bulunmadığı bir derlemde "Buzdolabında elma var", "Buzdolabında portakal var" cümleleri tamamen geçerli iken "Buzdolabında papaya var" cümlesi bilinmeyen

kelime içermektedir. Temsil modelini Türkçeyi yeni öğrenen bir öğrenci gibi düşünürsek; "Buzdolabında ***papaya*** var" cümlesini yerine "Buzdolabında ***bişey*** var", "Buzdolabında ***kinoa*** var" yerine "Buzdolabında ***bişey*** var" diyebilir. Glove ve Word2vec benzer bir yaklaşımla "bişey" kelimesi işlevi gören OOV (out-of-vocabulary) terimi için bir vektör rezerve etmektedir. Görüleceği üzere, Glove ve Word2vec yöntemlerinde derlemde yer almayan kelimelerin semantik ve işlevsel özelliklerinin yeterince öğrenilmesi mümkün olmamaktadır. Bu çok önemli bir dezavantajdır. Çünkü eğitim verisinde yer almayan kelimeler her koşulda olacaktır. Ayrıca, düşük frekansa sahip kelimeler de OOV kelimeleri gibi iyi temsil edilmemektedir. FastText ile bu probleme karşı etkili bir çözüm sunulmuştur. Kelimeler alt birim gömmelere (subword-level embeddings) ayrılarak n-gram kelime torbaları halinde temsil edilmektedir. Kelimenin alt birimlerinden oluşan her bir n-gram bir arada toplanarak kelime gömmesini oluşturur. Bu şekilde, kelime temsilinin sadece bir arada bulunduğu diğer kelimelerden yola çıkarak elde edilmesi yerine morfolojik yapısına ait özellikler de dahil edilmektedir. Karşılaşılmayan kelimelerin daha önce karşılaşılan harf gruplarından elde edilebilmesiyle temsil edilmesine olanak tanıyarak önemli bir avantaj sağlamaktadır. Örneğin n=3 ve n=4 için "papaya" kelimesi "pap", "apa", "pay", "aya", "papa", "apay", "paya" 3-gram ve 4-gram'ların toplamı şeklinde ifade edilir.

2.7. Diğer Hazır Kelime Gömmeleri

Önceden-Eğitilmiş Kelime Gömmeleri (Pre-Trained Word Embeddings — PWE), büyük metinler üzerinde eğitilen genel amaçlı kelime vektörleridir. Önceden eğitilmiş kelime gömmelerinin kullanılması; metin sınıflandırma, kümeleme, varlık tanıma gibi metin işleme görevlerinin başarısını iyileştirmeleri bakımından bir tür öğrenme aktarımı olarak kabul edilir. Derin öğrenme yöntemlerinin kullanıldığı DDİ problemlerinin çözümünde giriş verilerinin özellik çıkartımı konusunda sıklıkla tercih edilirler. Word2vec ve Glove en iyi bilinen kelime gömme algoritmalarıdır. Literatürde bunların dışında birçok PWE yöntemi mevcuttur ve zengin derlemler üzerinde eğitilerek kullanıma hazır bir şekilde sunulmaktadır. Bu tez çalışmasında önerilen doküman tabanlı kullanıcı temsili yöntemlerinde, duygu analizinde ve metin sınıflandırma çalışmalarında farklı PWE'lerden yararlanılmıştır.

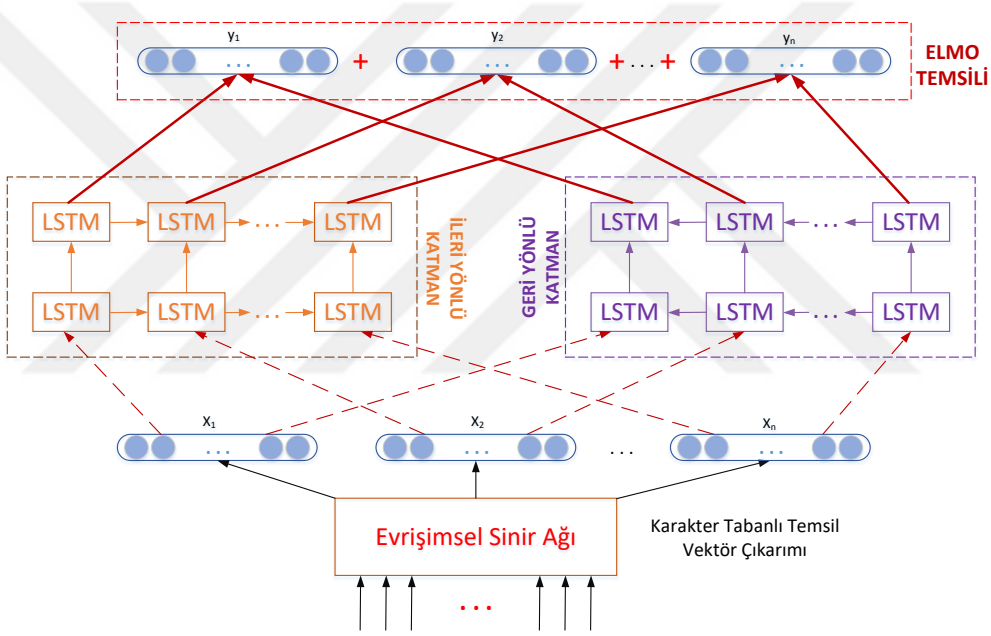
ConceptNet Numberbatch, çok dilli (multilingual) bir kelime temsil yöntemidir [69]. Mevcut yöntemler (Glove, Word2vec vb.) ile elde edilmiş kelime vektörlerinin çizge tabanlı yeni bir amaç fonksiyonu üzerinden iyileştirilmesine dayanmaktadır. Bunun için semantik bilgilerin gömüldüğü etiketleri ve ağırlıkları olan bilgi çizgelerini (knowlege graphs) kullanılmaktadır [70].

Google News, Google'nin sunduğu, yaklaşık 100 milyar kelime içeren bir derlem üzerinde eğitilmiş, 3 milyon tekil kelimeye ait 300 boyutlu vektörler içeren bir Word2vec modelidir.

Flair, metin sınıflandırma gibi çeşitli DDİ görevleri için dil modellerinin kolay bir şekilde kullanımını sağlayan bir kütüphanedir [37]. Aynı zamanda, ileri ve geri yönde eğitilen 2 farklı modelin birleştirilmesinden oluşan kendi isminde bir kelime gömme modeli sunmaktadır.

2.8. ELMO

Açılımı "Embeddings from Language Models" olan ELMO, dil modellerinden bağlamsal metin temsili elde edilmesi fikriyle ortaya çıkan derin öğrenme tabanlı bir yöntemdir [71]. Yapısını temel olarak iki yönlü Uzun-Kısa Süreli Bellek (Long Short-Term Memory - LSTM) ve Evrişimli Sinir Ağları (Convolutional Neural Networks - CNN) modelleri oluşturmaktadır. ELMO modelinin temel çalışma prensibi Şekil 2.19'da verilmiştir.



Şekil 2.19. ELMO temel çalışma prensibi

Bu yöntemde giriş metni modele karakter tabanlı olarak aktarılır. Her karakter bir vektördür ve dil modeline verilen ara kelime temsilleri bu vektörlerin CNN modelinden geçirilmesiyle elde edilir. ELMO'yu oluşturan LSTM katmanlar İki-Yönlü Derin Dil Modeli (Deep Bidirectional Language Model – biLM) görevi görmektedir. Giriş karakterlerinin doğrudan biLM'ye aktarılması yerine CNN ile ara temsillerin elde edilmesi sayesinde dile ait zamansal (temporal) bilgilerle birlikte n-gram gram özellikleri de öğrenilerek güçlü bir dil modelleme süreci uygulanır. Karakter evrişimlerinin (convolutions) bir diğer avantajı ise eğitim verisinde yer almayan OOV terimlerin daha sonra karakter temsilleri üzerinden etkili bir şekilde temsillerinin elde edilmesine olanak sağlamasıdır.

2.9. BERT

Açılımı “Bidirectional Encoder Representations from Transformers” olan BERT (İki-yönlü dönüştürücü enkoder), Google tarafından geliştirilmiştir [72]. En önemli özelliği ELMO gibi bağlama duyarlı (Context-aware) bir yaklaşım olmasıdır. Dilin biçimbilimsel özelliklerini öğrenmesi ve bağlama duyarlı tahmin yürütebilmesi bakımından Türkçe metinler üzerinde gerçekleştirilen çalışmalarda kullanımının giderek popülerleştiği görülmektedir [73,74].

Glove, Word2vec ve FastText yöntemleri kelime sırasını hesaba katmamaktadır ve bağlama duyarlı değildirler. Bu modeller eğitim kümesi sözlüğünde yer alan her kelimeye karşılık birer vektör değeri üretir. Eğitim sonrasında kaydedilen kelime vektörleri kullanılır. Yeniden eğitim veya iyileştirme gibi amaçlar haricinde modellere ihtiyaç kalmaz. Bağlama duyarlı modellerde ise kelimenin vektörel karşılığı anlık olarak birlikte bulunduğu çevreleyen kelimelere bağlı olarak her seferinde yeniden üretilir. Örneğin, “Batman Belediyesi, kentin ismini izinsiz kullanması sebebiyle 'Batman' filminin yapımcısı Amerikan DC Comics şirketini dava edebileceğini açıkladı” cümlesinde “'Batman” iki farklı anlama sahiptir. Birincisi şehir adı olan Batman’dır. İkincisi ise “Kara Şövalye” gibi filmlerde yer alan kurgusal süper kahramanın adıdır. Statik bir kelime gösterimi yönteminde, Batman kelimesi karşılığında yalnızca bir vektör bulunmaktadır. Dinamik bir temsil sisteminde ise kelime vektörü bağlama bağlı olarak üretilir ve sabit değildir.

BERT modeli, dönüştürücü adı verilen yeni bir sinir mimarisine sahiptir [71]. Giriş metninin alt kelime belirteçlerini (sub-word tokens) kullanır ve bu simgelerden sözcükler oluşturur. Ardından, sonraki cümle tahmini ve maskelenmiş (masked) dil modeli olmak üzere iki denetimsiz görev için büyük miktarda veriler üzerinde ön eğitim (pre-training) gerçekleştirir. Ön eğitim tamamlandıktan sonra metin özetleme, sınıflandırma, duygu analizi gibi çeşitli DDİ görevlerinde kullanılır.

3. ÇOK KAYNAKLI, DENGESİZ VERİ SETİ ÜZERİNDE DERİN ÖĞRENME YÖNTEMLERİYLE DUYGU ANALİZİ

3.1. Problemin Tanımı

Duygu analizi kısaca; bir görüş idafe eden metnin pozitif mi yoksa negatif mi olduğunun bulunmasıdır. Çevirimiçi yorumların duygu analizi, oldukça ilgi gören bir çalışma alanı haline gelmiştir. Son yıllarda yaygın bir şekilde DDİ problemlerine uygulanan derin öğrenme metotlarının bu alanda da geleneksel makine öğrenmesi yöntemlerinin yerini aldığı görülmektedir.

Pang ve Lee [75], duygu analizi ve fikir madenciliği konularını detaylı olarak ele aldıkları çalışmada ticari ürün, otel, kitap, restoran, seyahat firması, doktor tercihi ve hayatın birçok farklı alanında insanların başkalarının görüşlerini merak ettiğini ve bir konuda "diğer insanlar ne düşünüyor ?" sorusuna cevap aramanın bir çok karar alma sürecine etki eden önemli bir bilgi edinme problemi olduğunu ortaya koymaktadır.

Sosyal medya ve e-ticaret gibi günlük hayat aktivitelerinin çevrimiçi ortamlara taşınmasına olanak sağlayan teknolojik gelişmeler, yapısal olan ve yapısal olmayan (structured/unstructured) metinler üzerinde kullanıcı görüşlerinin olumlu (pozitif), olumsuz (negatif) ve tarafsız (nötr) gibi kategorilerden hangisini içerdiğinin tespit edilmesi amacıyla gerçekleştirilen duygu analizi çalışmalarının önemini giderek arttırmaktadır. DDİ alanının popüler çalışma konularından birisi olan duygu analizi, 2000'li yıllarla birlikte veri madenciliği, web madenciliği, metin madenciliği alanlarında da karşımıza çıkmaya başlamıştır ve artık ticari ve sosyal hayat içerisinde artan öneminden dolayı sosyal bilimler ve yönetim bilimleri gibi bilgisayar bilimleri dışındaki disiplinlerin de bir konusu hâline gelmiştir [76].

Literatürdeki duygu analizi çalışmaları incelendiğinde baskın olarak İngilizce metinlerin kullanıldığı ve dolayısıyla mevcut veri setlerinin çoğunluğunun İngilizce olduğu görülmektedir [77]. Türkçe veri setleri üzerinde duygu analizi çalışmaları artan bir popülerlikle devam etmekle birlikte bu çalışmalarda kullanılan veri setleri genellikle çalışma özelinde oluşturulmuştur. Türkçe için kıyaslama (benchmark) amaçlı, herkese açık veri setlerinin oluşturulması önemlidir [78].

Bu çalışmada Türkiye'deki çeşitli popüler web sitelerinden (Beyazperde, N11, Tripadvisor ve Kitapyurdu) kullanıcı yorumları toplanıp bir araya getirilerek zengin bir Türkçe duygu veri seti elde edilmiştir. ***Bigailab-2sentiment-2M*** ismi verilen bu veri seti üzerinde derin öğrenme yöntemleri uygulanarak yüksek doğruluklu duygu sınıflandırma modellerinin geliştirilmesi hedeflenmiştir.

Çalışmanın motivasyonunu, mevcut çalışmaların incelenmesi sonucunda gerçek hayat kullanımları için etkili bir uygulama geliştirmek konusunda yeterince dikkat çekilmediğine kanaat getirilen bazı önemli koşullar meydana getirmiştir. Bu koşullar şunlardır:

- Büyük miktarda veri (milyonlarca kullanıcı yorumu) kullanılması
- Türkçe veriler üzerinde gerçekleştirilmesi
- Verilerin farklı kaynaklardan elde edilerek bir araya getirilmesi
- Sınıf dengesizliği

Bu koşullar çeşitli çalışmalarda ele alınsa da, hepsinin bir arada dikkate alındığı bir senaryo için aşağıda maddeler hâlinde sıralanan katkılara ihtiyaç duyulduğu düşünülmüştür:

- ✓ Klasik derin öğrenme yöntemlerinin yukarıda verilen şartlarda nasıl performans gösterdiğinin ortaya konulması.
- ✓ Klasik yöntemlerin başarılı modellerinden yola çıkarak daha karmaşık hibrit modellerin önerilmesi.
- ✓ Farklı modeller karşılaştırıldığı için başarı ölçümlerinin çeşitli şans faktörlerinden etkilenmeden adil bir şekilde yapılması. Bunun için başlangıç ağırlıklarının yüklenmesi, verilerin eğitim ve test olarak bölünmesi aşamalarındaki rasgeleliğin karşılaştırılan modeller arasında sabit tutulması ve K-Fold çapraz doğrulama yönteminin kullanılması.
- ✓ Sınıf dengesizliği probleminin çözümüne yönelik öneriler geliştirilmesi ve bunların farklı modeller için karşılaştırmalı olarak ne kadar iyi performans gösterdiğinin ortaya konulması.

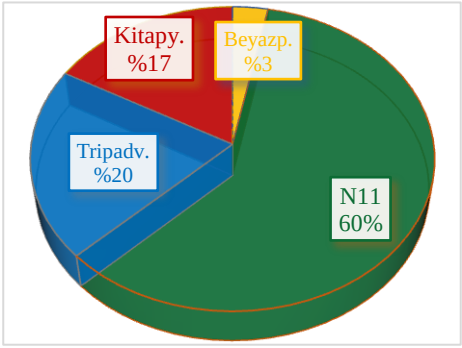
3.2. Veri Setinin Oluşturulması ve Analizi

Bu çalışmadaki veriler, çeşitli internet sitelerinden, kullanıcıların belirli bir zaman aralığında paylaştığı yorumların otomatik olarak toplanmasıyla elde edilmiştir. 2 milyonun üzerinde yoruma ait sayısal bilgiler Tablo 3.1'de verilmiştir.

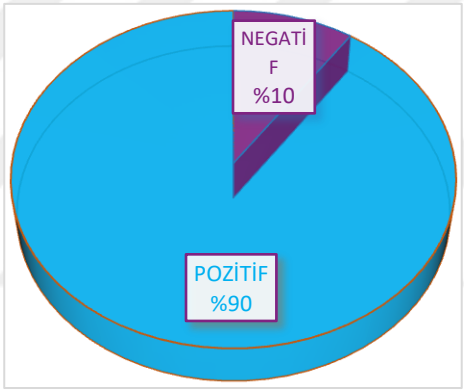
Tablo 3.1. Veri seti genel bilgileri

Kaynak	Açıklama	Negatif	Pozitif	Toplam
Beyazperde	Film yorumları	18.763	44.070	62.833
N11	E-ticaret (genel)	109.611	1.094.136	1.203.747
Tripadvisor	Seyahat tavsiyesi	53.415	355943	409.358
Kitapyurdu	E-ticaret (kitap)	12.914	326300	339.214
	Genel toplam	194.703	1.820.449	2.015.152

Verilerin toplandığı web sitelerine göre dağılımları Şekil 3.1'de, bütün veriler içerisindeki pozitif ve negatif yorumların yüzdelik oranları ise Şekil 3.2'de grafik olarak verilmiştir.



Şekil 3.1. Duygu veri setindeki farklı kaynakların dağılımı



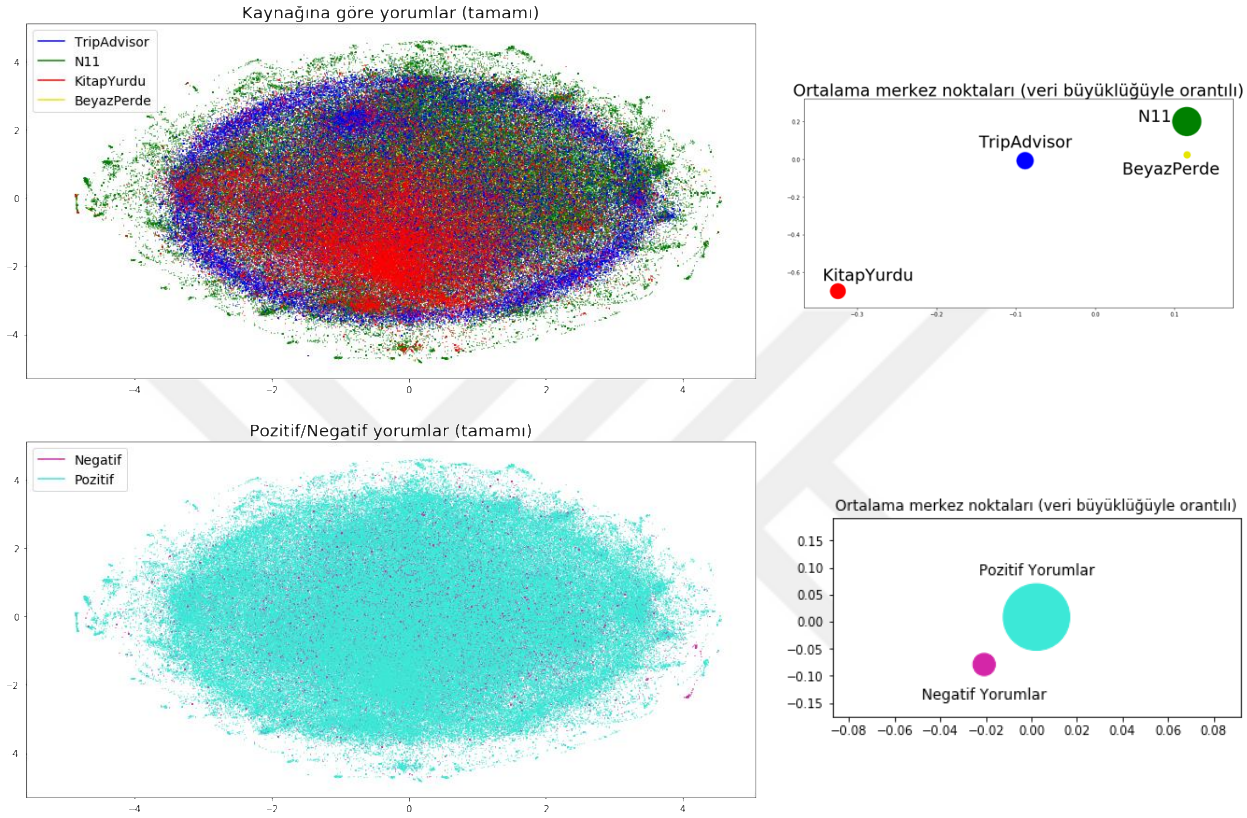
Şekil 3.2. Duygu veri setindeki pozitif ve negatif veri dağılımı

Sınıf etiketleri belirlenirken her bir web sitesinin kendi oluşturduğu puanlama sistemi göz önünde bulundurulmuştur. Kullanıcılar, yorum yazmaya ek olarak yıldız/puan verme ile beğeni derecesini gösteren bir oylama yapmaktadır. Örneğin Beyazperde sitesinde filmler için yapılan yorumlar 10 farklı puandan (0,5 – 1,0 – 1,5 … 4,5 – 5); Kitapyurdu web sitesinde ise kitaplar için yapılan yorumlar 5 farklı puandan (1 – 2 – 3 – 4 – 5) oluşmaktadır. Yüksek puanlı yorumların pozitif, düşük puanlı yorumların ise negatif duygu belirttiği kabul edilmiştir. Ara değerde olan diğer yorumlar veri setine dâhil edilmemiştir.

3.2.1. Veri Setinin Görselleştirilmesi

Veri setinin, verilerin kaynağına ve etiketlerine göre nasıl bir dağılım gösterdiğinin analizi yapılmıştır. Bu analiz için öncelikle her bir yorumun 100 boyutlu dağıtık doküman temsil vektörü elde edilmiştir. Daha sonra, bu vektörlerin 2 boyutlu uzayda karşılık geldiği noktalar elde

edilmiştir. Doküman temsilinde *doc2vec* modeli, 2 boyuta indirgeme işleminde *t-SNE* görselleştirme yöntemi kullanılmıştır. Aynı zamanda Bölüm 5 ve Bölüm 6'da da kullanıcı temsillerinin görselleştirilmesinde yararlanılan bu tekniklerle ilgili açıklamalara bu bölümde ayrıca yer verilmemiştir.



Şekil 3.3. Yorumların doküman vektörlerinin görselleştirilmesi

Şekil 3.3'te 4 farklı veri görselleştirme işleminin sonucu bir arada sunulmuştur. Yorumlar, elde edildikleri web sitesi baz alınarak renklendirilmiştir ve veri-kaynağına-göre dağılımlarını gösteren bir çizim elde edilmiştir. Çok geniş bir ürün yelpazesine sahip olan N11 alışveriş sitesinden elde edilen ve yeşil renkle gösterilen yorumların 2 boyutlu temsili olan noktaların belirli bir alanda yoğunlaşmadığı görülmektedir. Sadece kitap satışı üzerine hizmet veren Kitapyurdu web sayfasının yorumlarını temsil eden kırmızı renkli noktaların ise diğer veri kaynaklarına göre daha çok bir araya toplanma eğilimi gösterdiği anlaşılmaktadır. Veri setindeki tüm yorumların ancak yaklaşık %3'üne sahip olan Beyazperde kaynağına ait veriler (sarı renkli) ise çok az miktarda olduğu için fark edilir derecede görülememektedir. Her ne kadar seyahat ve gezi başlığında bir araya getirilmiş işletmeler hakkında yorumlara sahip olsa da; işletmeler arasında yemek, konaklama, aktivite danışmanlığı gibi oldukça geniş bir hizmet alanı olan Tripadvisor yorumlarının temsilleri (mavi renkli) de hemen hemen her bölgeye dağılmıştır.

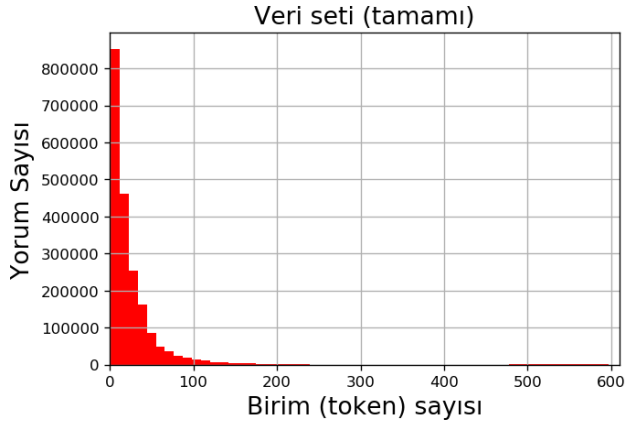
Şekil 3.3'te, veri kaynağına göre renklendirilen noktalar karmaşık bir görüntü oluşturmuştur. Bu çizimin daha kolay incelenebilmesi için, her bir kaynağa ait noktalar yatay ve düşey eksen değerlerinin ortalamasına denk gelen konumlarda ayrıca çizdirilmiştir. Nokta büyüklükleri, ilgili renge ait yorum sayısıyla orantılı olarak ayarlanmıştır.

Şekil 3.3'te yer alan diğer iki çizim ise benzer bir incelemenin verilerin etiketleri üzerinden yapılmasına olanak vermektedir. Yorumlar, pozitif ve negatif olmalarına göre ortalamaları üzerinden bir miktar ayrışma gösterse de tekil olarak ele alındığında oldukça karmaşık bir dağılım göstermektedir. Dolayısıyla, veri setinin görselleştirilmesi sonucunda yorumların hem elde edildiği kaynak hem de ait olduğu sınıf bakımından dengesiz bir dağılıma sahip olduğu görülmüştür.

3.2.2. Metin Uzunluğu Parametresinin Belirlenmesi

Çalışmanın ana amacı, dengesiz veri seti üzerinde duygu analizi gerçekleştirmektir. Dengesiz veri problemlerinde en temel yaklaşımlar; az olan verinin çoğaltılması veya fazla olan verinin sınırlandırılmasına dayanır. Çalışmada, farklı web sitelerinden, farklı sayılarda elde edilen yorumlardan oluşan bir veri seti kullanılmıştır. Bundan dolayı, belirlenen parametrelerin herhangi bir sınıfa veya kaynağa yönelik taraflılık (bias) oluşturmamasına dikkat edilmesi gerekmektedir.

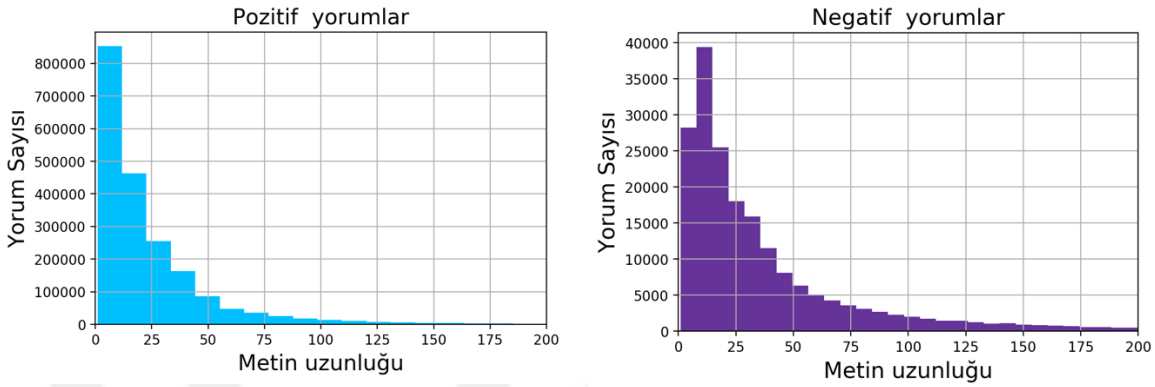
Metin uzunluğu üzerinden yapılan veri incelemesi dikkate alınarak derin öğrenme modelleri için belirlenmesi gereken metin uzunluğu parametresinin seçilmesi amaçlanmıştır. Veri seti genelinde, kaynak web sitesi bazında ve etiket türüne göre yorumların içerdiği karakter veya kelime sayısı bakımından bir örüntü olup olmadığı araştırılarak detaylı bir inceleme yapılmıştır.



Şekil 3.4. Duygu veri seti - metin uzunluğu histogramları

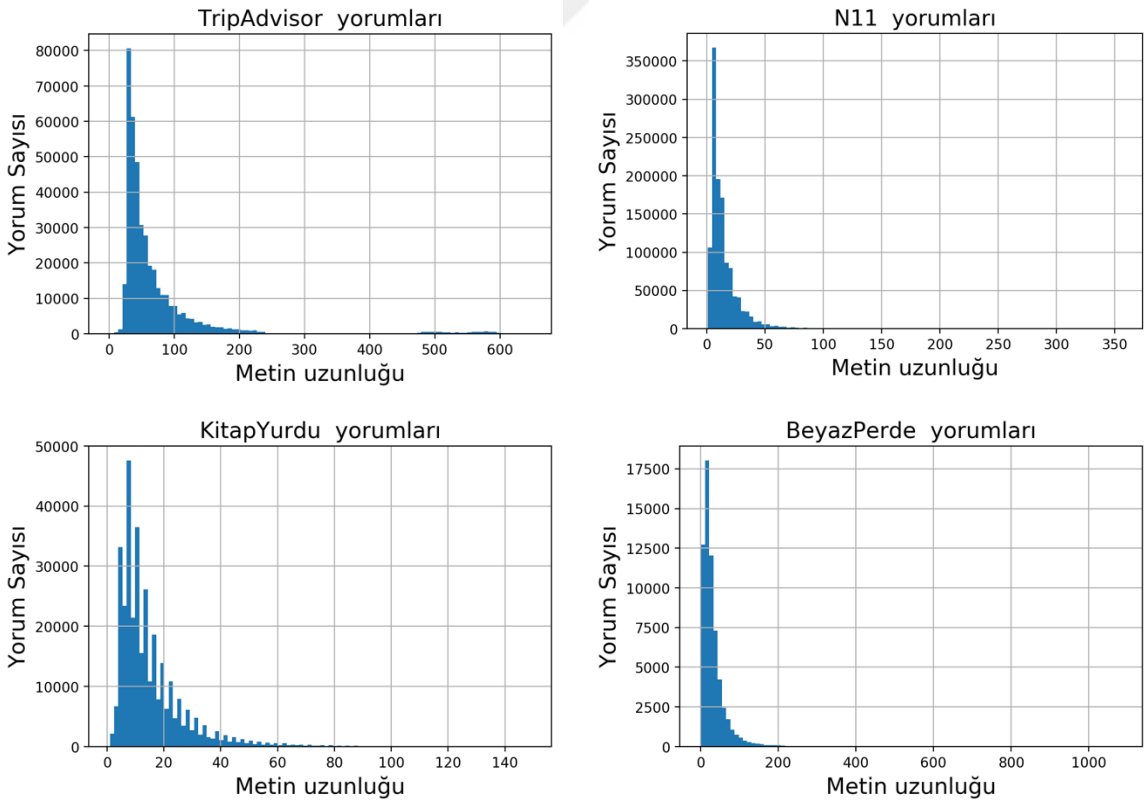
Veri setindeki yorumların içerisindeki boşluklara göre parçalanmasıyla elde edilen birim (token) sayısına bakıldığında, bir yorumda ortalama olarak 27, en uzun yorumda 1084 adet birim

bulunduğu görülmektedir. Şekil 3.4'te veri seti geneli için token sayıları histogramı görülmektedir. Yorumların pozitif ve negatif etiketli olmalarına göre kelime sayısı histogramları Şekil 3.5'te verilmiştir. Bu bölümdeki histogramların kutu sayısı (bins) 100 olarak ayarlanmıştır.



Şekil 3.5. Duygu veri seti - etikete göre kelime sayısı histogramları

Metin uzunlukları, farklı web sitelerine ait yorumlar için ayrı ayrı histogram olarak elde edilmiştir ve Şekil 3.6'da verilmiştir.



Şekil 3.6. Veri kaynaklarına göre kelime sayısı histogramları

Metinlerin karakter sayısı bakımından uzunlukları ise veri kaynağına ve türüne göre Tablo 3.2'de detaylı olarak verilmiştir. Tablo 3.2 incelendiğinde veri seti genelinde negatif yorumların pozitif yorumlardan yaklaşık 2 kat daha uzun olduğu, fakat en uzun yorumların pozitif olduğu görülmektedir.

Tablo 3.2. Yorumların metin uzunlukları

Veri kaynağı/türü	Karakter sayısı aralığı (min – max)	Ortalama	Standart sapma
Beyazperde	3 - 7740	240	290.9
N11	2 - 2305	103	97.66
Tripadvisor	27 - 4951	524	598.5
Kitapyurdu	8 - 1001	115	102.1
Pozitif	2 - 7740	178	287.5
Negatif	2 - 4980	343	593.7
Hepsi	2 - 7740	195	333.3

Sonuç olarak maksimum metin uzunluğu parametresinin 100 olarak seçilmesine karar verilmiştir. Bu değer, veri seti tamamının %96,61'ini doğrudan kapsamaktadır. Aynı zamanda negatif yorumların %89,61'ini ve pozitif yorumların %97,36'sını kapsamaktadır. Özetle, toplam 68.229 adet yorumun ilk 100 kelimeden sonrası model eğitimine dâhil edilmemiştir. 1.946.923 yorum ise kesintiye uğramadan kullanılmıştır.

3.3. Metin Önişleme Adımları ve Kelime Gömme Katmanı

Metin, yapısal olmayan bir veri türüdür. Kullanıcı yorumlarından oluşan metin veri setinin öncelikle derin öğrenme algoritması girişinde temsil edilmek üzere öz niteliklerinin belirlenmesi gerekir. Öz nitelik olarak karakter veya sözcük tabanlı metin temsil yöntemleri kullanılabilir. Bu çalışmada sözcük tabanlı temsillerin kullanılması tercih edilmiştir. Önişleme adımlarında ilk olarak metinler küçük harflere dönüştürülmüştür. Noktalama işareti, sekme (tab), satır boşluğu gibi her türlü alfa numerik olmayan karakterlerden arındırılmıştır. Fakat durak kelimelerin (stop-words) temizlenmesi ve kelime köklerinin bulunması gibi detaylı metin önişleme tekniklerinin uygulanması gerekli görülmemiştir.

Metinler boşluklar üzerinden sözcüklere ayrılmıştır ve veri setinin en yüksek frekanslı 10.000 sözcüğü belirlenmiştir. Daha düşük frekanslı olmasından dolayı bu listeye giremeyen diğer bütün sözcüklerin yerine kullanılmak üzere "UNK" sözcüğü kelime listesine eklenerek

toplam 10000+1 sözcükten oluşan bir derlem sözlüğü oluşturulmuştur. Sözlükteki her kelime bir tam sayı değeri ile eşleştirilerek sözlük-dizin (index) yapısı oluşturulmuştur. Yorumlar derin öğrenme modeline verildiğinde kelime gömme katmanı (word embedding layer) öncesinde bu dizin değerleriyle temsil edilmektedir. Dizin değerlerinin ilki, yani sıfır indisli elemanı ise rezerve edilmiştir. Şekil 3.7'de örnek 4 metin için dizin değerlerinin dönüşümü gösterilmiştir. Her bir terimin ayrı bir sayısal değere karşılık geldiği dönüştürme işleminde sözlükte yer almayan diğer bütün kelimeler için "UNK" olarak ayrılan sözcüğün indisi olan 10.000 değerinin verildiği görülmektedir.

```
yorumlar = [['herkese', 'güzel', 'bir', 'yaşam',  'diliyorum'],
            ['herkese', 'sağlıklı', 'bir', 'yaşam',  'diliyorum'],
            ['herkese', 'sıhhatli','bir', 'yaşam',  'diliyorum'],
            ['x0zm', 'x1zm', 'x2zm', 'x3zm', 'x4zm']]

terim_listesi = tokenizer.texts_to_sequences(yorumlar)

for yorum in terim_listesi:
    print(yorum)
```

```
[104, 6, 3, 2829, 2414]
[104, 1899, 3, 2829, 2414]
[104, 10000, 3, 2829, 2414]
[10000, 10000, 10000, 10000, 10000]
```

Şekil 3.7. Sözlük-dizin değerleri dönüşümü örneği

```
yorumlar = [['ürünü', 'çok', 'beğendim', 'tekrar', 'teşekkürler'],
            ['güzel', 'bir','otel', 'teşekkürler'],
            ['harika', 'bir', 'kitap'],
            ['film', 'kötüydü'],
            ['berbat']]

terim_listesi = tokenizer.texts_to_sequences(yorumlar)
pad_sequences(terim_listesi, maxlen=10001)
```

```
array([[    0,     0,     0, ...,    55,    82,    19],
       [    0,     0,     0, ...,     3,    11,    19],
       [    0,     0,     0, ...,    31,     3,    35],
       [    0,     0,     0, ...,     0,   123,  1493],
       [    0,     0,     0, ...,     0,     0,   237]], dtype=int32)
```

Şekil 3.8. Maksimum metin uzunluğunun ayarlanması örneği

Derin öğrenme modellerinin giriş katmanına aynı boyutlarda elemanlardan oluşan eğitim verilerinin uygulanması gerekmektedir. Bu elemanların sabit boyutlu olabilmesi için bir yorumun oluşturan maksimum terim sayısı 100 olarak belirlenmiştir. Maksimum terim sayısını aşan yorumların sadece ilk 100 terimi alınarak fazla sözcükler elenmiştir. Üst sınırdan daha az sayıda terimden oluşan yorumlara ise 100 terime tamamlanacak şekilde ekleme yapılmıştır. Ekleme

işleminde (padding) daha önce rezerve edilmiş olan sıfır indisi kullanılmıştır. Şekil 3.8'de farklı uzunluktaki yorumların aynı uzunluktaki giriş vektörlerine dönüştürülmesi ve eksik terimlerin sıfırlarla doldurulması gösterilmektedir.

Derin öğrenme modeli girişi katmanında yorumların terim sayısı bakımından eşit uzunluklu olması gerektiği gibi her kelimeyi temsil eden kelime gömmelerinin de önceden belirlenen sabit bir uzunlukta olması gerekmektedir. Bu çalışmada Türkçe için geliştirilen FastText [79] dil modelinin 300 boyutlu hazır kelime gömmeleri kullanılmıştır. 10001 x 300 boyutlu bir matris oluşturularak bütün ilk değerler sıfır olacak şekilde atanmıştır. Daha sonra her bir terimin denk geldiği matris satırı dil modelindeki ilgili kelimeye ait vektörle değiştirilmiştir. Veri setindeki en yüksek frekansa sahip ilk 10000 terimin 9057'si FastText dil modelinde de yer almaktadır. Kalan 943 kelimenin, "UNK" teriminin ve sıfır indisine karşılık vektörün başlangıç değeri ise sıfır olarak bırakılmıştır. Kelime gömme katmanının giriş ve çıkışlarının yapısı Şekil 3.9'da gösterilmiştir.

Şekil 3.9. Kelime gömme katmanı

Bu çalışmada kullanılan bütün derin öğrenme modellerinde ortak olarak Şekil 3.9'da gösterilen giriş ve kelime gömme katmanları kullanılmıştır. Modellerin eğitimi sırasında bu katman güncellenebilir olarak ayarlanmıştır. Dolayısıyla eğitim tamamlandığında aynı dokümanı temsil eden matrisler başlangıçtan farklı olacaktır.

3.4. Derin Öğrenme Modelleri

Yapay sinir ağları temeline dayanan derin öğrenme, son 15 yılda giderek popülerleşmekte olan bir makine öğrenmesi tekniğidir. Standart bir YSA; giriş katmanı, bir veya daha fazla gizli katman ve çıkış katmanından oluşmaktadır. Derin öğrenme yaklaşımında ise giriş katmanından amaç fonksiyonunun değerlendirilmesine kadar geçen süreçte araya yerleştirilen bir dizi katmanla veriden kendi kendine öğrenebilen çok daha fazla parametreye sahip modeller kullanılmaktadır. Bu derin mimarileri oluşturan katmanlar, probleme uygun olarak bir araya getirildiği takdirde veriye ait özellikler hiyerarşik bir şekilde öğrenilir [80].

İlk dikkat çekici başarılarını bilgisayar görüsü problemleri üzerinde gösteren ResNet [81], AlexNet [82], VggNet [83] gibi derin sinir ağlarından (Deep Neural Network – DNN) oluşan mimariler nesne tespiti, sınıflandırma ve segmentasyon gibi bilgisayar görüsü görevlerinde

yüksek performans göstermiştir. Bu modeller sonuç itibariyle birer "net" yani sinir ağı olsa da standart YSA'lara göre daha karmaşık ve veriye ait önemli ayırt edici özelliklerin otomatik olarak öğrenilmesi konusunda çok daha başarılıdır.

Metin sınıflandırma, varlık ismi tanıma, makine çevirisi, soru cevaplama gibi çeşitli DDİ problemlerinde yüksek başarı gösteren DNN modellerin diğer makine öğrenmesi yöntemlerine kıyasla en önemli avantajlarından birisi çıkış katmanından elde edilen ana probleme ait hata bilgisinin geriye yayılım algoritmasıyla önceki katmanlara iletilmesinin bir yandan giriş katmanındaki dağıtık metin temsillerinin daha iyi öğrenilmesini sağlayan parametre güncellemelerini tetiklemesidir. En önemli avantajlarından bir diğeri ise kelime temsilleri gibi DDİ problemlerinin en önemli veri özelliklerinin elde edildiği vektörlerin yüksek miktarda zengin veri kaynaklarından (Wikipedia, sosyal medya, gazeteler, kitaplar …) elde edilen derlemler üzerinde eğitildikten sonra istenilen problem içerisinde kullanılabilmesi ve bir önceki özellikte belirtildiği şekilde bir yandan problem içerisinde gösterilen başarıya göre (task-oriented) iyileştirme (fine-tuning) yapılmasının mümkün olmasıdır.

CNN, Tekrarlayan Sinir Ağları (Recurrent Neural Networks – RNN) ve RNN'in gelişmiş versiyonları olan LSTM, İki-Yönlü LSTM (bidirectional LSTM / Bi-LSTM) ve Geçitli Tekrarlayan Birim (Gated Recurrent Unit - GRU) metin sınıflandırma problemlerinde yaygın olarak kullanılan derin öğrenme metotlarıdır. Bu çalışmada Çok Katmanlı Algılayıcılar (Multi Layer Perceptron - MLP), CNN, LSTM ve BiLSTM modelleri ile bu modellerin çeşitli şekillerde bir araya getirildiği hibrit derin öğrenme modelleri kullanılarak duygu analizi veri setinin pozitif ve negatif yorumlarının sınıflandırılması gerçekleştirilmiştir.

Modellerin implementasyonunda Keras [84] kullanılmıştır. Google tarafından geliştirilen açık kaynaklı makine öğrenmesi yazılım kütüphanesi olan TensorFlow [85] üzerinde çalışan, Python dilinde yazılımış bir API olan Keras, derin öğrenme modellerinin mimarilerinin yazılımında yaygın olarak kullanılmaktadır.

3.4.1. MLP

İleri yönlü ve tam bağlı bir yapıya sahip, FFNN'nin bir çeşidi olan MLP'nin geçmişi 1960'lara kadar dayanmaktadır. Basit bir yapıya sahip olan MLP birçok örüntü tanıma probleminde kullanılmaktadır. Günümüzde yerini çok daha karmaşık derin ağlara bırakmış olsa da bir problemin derin öğrenme yöntemleriyle ele alınması sürecinde temel başarı göstergesi (baseline) olarak tercih edilmesinin çeşitli avantajları vardır. Örneğin, tekrarlayan bağlantılar veya konvolüsyon işlemleri içeren daha karmaşık derin öğrenme yöntemleri uygulanırken asgari olarak temel bir MLP modeline göre daha iyi bir performans göstermesi gerektiği fikrinden yola çıkarak, ortaya konulan yaklaşımın katkısı hakkında fikir sahibi olunabilir. Ayrıca özellik çıkarma gibi amaçlarla farklı özgün modellerin öğrenme aşamalarına dâhil edilebilirler [86].

Bu çalışmada yaygın derin öğrenme modellerinin performansları ve dengesiz veri problemine yönelik uygulanan iyileştirme yaklaşımının katkısı incelenirken MLP modeli de karşılaştırmalara dâhil edilmiştir. Kelime gömme katmanı, gömme katmanı sonucunu düzleştirici (flatten) katman ve 2 adet yoğun katmandan oluşan modelin detaylı parametreleri Şekil 3.10'da verilmiştir.

```
_model_name == "MLP":
_model = Sequential()
_model.add(embedding_layer)
_model.add(Flatten())
_model.add(Dense(128))
_model.add(Activation('relu'))
_model.add(Dropout(0.3))
_model.add(Dense(64))
_model.add(Activation('relu'))
_model.add(Dropout(0.2))
_model.add(Dense(num_of_classes))
_model.add(Activation('softmax'))
```

Şekil 3.10. MLP modeli özellikleri

Sınıflandırıcı olarak kullanılan 4 nöronlu yoğun katman dışındaki yoğun katmanların çıkışlarında ReLu aktivasyonu tercih edilmiştir. Bu seçim çalışmadaki diğer modeller için de aynı şekildedir.

3.4.2. CNN

Derin sinir ağların gradyan tabanlı yöntemlerle eğitilmesi ile ses ve görüntü gibi karmaşık verilerin lineer olmayan özelliklerinin öğrenilmesi konusunda kaydedilen en önemli aşamalardan birisi CNN'ler olmuştur [82,87]. CNN'ler, nesne tanıma problemlerinde gösterdiği başarılarla birlikte derin öğrenme yöntemlerinin temel tekniklerinden biri hâline gelmiştir. Otomatik soru cevaplama, metin özetleme ve dil modelleme gibi birçok DDİ uygulamalarında da yararlanılan CNN modelleri, özelliklerin aşamalar halinde öğrenilmesine olanak sağlayan bir mimariye sahiptir.

Giriş verisi, piksel değerlerinden oluşan görüntü matrisi veya metin gömme katmanı gibi temsiller aracılığıyla derin öğrenme ağına aktarılır. CNN modelinin ön önemli avantajlarından birisi bu katmanların sadece belirli bölgelerine bağlı gizli katman bağlantılarıyla verilere ait farklı özelliklerin keşfedilmesine odaklanabilmesidir. Filtre adı verilen bu bölgesel bağlantılar, kendisinden önce gelen katman üzerinde gezdirilir ve farklı verilerin aynı tür özellikleri elde edilir.

Geleneksel görüntü işlemede, nesne tanıma gibi problemlerde görüntünün kenarlarının (edge) veya sınırlarının (boundary) tespit edilmesine yönelik özel teknikler uygulanırken, CNN modelinde bu önemli özellikler otomatik olarak keşfedilir. Derin öğrenme yöntemleri ile geleneksel makine öğrenmesi yöntemleri arasındaki benzer yaklaşım farklılıkları metin işleme alanında da mevcuttur. Örneğin metin benzerliği, otomatik soru cevaplama, metin özetleme gibi problemlerde bir cümle içerisinde yer alan n-gramların tespit edilmesi önemli bir özellik çıkarımıdır. Artık eski olarak nitelendirilen yöntemlerde kelimelerin semantik anlamlarına bakılmaksızın bütün derlem içerisindeki frekansları üzerinden çeşitli seçimler (ağırlıklandırma vb.) yapılarak n-gramlar belirlenmektedir. CNN'de ise giriş metninin yerel bölgelerine uygulanan filtrelerin boyutları oranında n-gram'lardan yararlanılmaktadır. Bunun için özellikle kurgulanmış bir n-gram çıkarım aşaması veya katmanı yoktur. Fakat bu ve benzeri birçok semantik ve sözdizimsel özellikler model tarafından otomatik olarak keşfedebilmektedir [60,88].

CNN'ler, klasik çok katmanlı sinir ağlarından farklı olarak bir dizi evrişim ve ortaklama (pool) katmanı içerir. Ortaklama ile katmanların daha önemli özelliklerinin tutulması ve sonraki katmanlara daha küçük boyutlarda aktarılması (downsampling) sağlanır. Bu çalışmada önerilen CNN modelinin katmanlarının ve önemli parametrelerinin detayları Şekil 3.11'de verilmiştir.

```
_model_name == "CNN":
_model = Sequential()
_model.add(embedding_layer)
_model.add(Conv1D(filter_num = 128,  kernel = 5, activation='relu'))
_model.add(MaxPooling1D(pool = 3))
_model.add(Conv1D(filter_num = 128,  kernel = 5, activation='relu'))
_model.add(MaxPooling1D(pool = 3))
_model.add(Conv1D(filter_num = 128,  kernel = 5, activation='relu'))
_model.add(GlobalMaxPooling1D())
_model.add(Dense(filter_num = 128))
_model.add(Activation("relu"))
_model.add(Dropout(0.3))
_model.add(Dense(num_of_classes))
_model.add(Activation('softmax'))
```

Şekil 3.11. CNN modeli özellikleri

Evrişim ve ortaklama adımlarının üçer kez arka arkaya uygulanmasının ardından 128 nöronlu bir yoğun katmanla devam eden CNN modeli sınıflandırma işlemini gerçekleştiren 4 nöronlu çıkış katmanıyla sonlandırılmıştır.

Kernel boyutu, filtre sayısı, aktivasyon fonksiyonu ve ortaklama katmanı tercihi CNN modelinin başarısını etkileyen önemli parametrelerdir. Çalışmanın kapsamını amacının üzerinde genişletmemek üzere farklı CNN model mimarileri ve parametre araştırmalarının etkilerine yönelik en iyileme süreçleri tez içerisinde paylaşılmamıştır. Gerçekleştirilen farklı deneylerde

elde edilen sonuçlar gözetilerek model genelinde filtre sayısı 128, kernel boyutu 5x5 ve ortaklama katmanı boyutu 3x3 olarak seçilmiştir.

3.4.3. RNN, LSTM ve Bi-LSTM

RNN'ler, verilerin sıralı olduğu (sequential) problemlerde geleneksel ileri beslemeli sinir ağlarına göre daha etkili olma düşüncesiyle geliştirilmiş derin öğrenme modelleridir. RNN ve gelişmiş türevleri olan LSTM varyantlarında veriler kelimelerin belirli bir sırayla art arda gelerek cümle oluşturması gibi diziler hâlinde işlenir. Önceki girişlerin model çıktısına zaman üzerinde zincirleme bir şekilde etki etmesi için tekrarlayan bir mekanizmaya sahiptir [46].

RNN varyantlarından metin üretimi (text generation), konuşma tanıma, makine çevirisi, duygu analizi gibi çeşitli DDİ konularında yararlanılmaktadır. Sesten metne dönüştürme (speech-to-text), makine çevirisi gibi problemlerde model çıktısı yine bir dizidir. Buna diziden-diziye (sequence-to-sequence) öğrenme denir. Bu çalışmada giriş metninin sonucunda pozitif/negatif sınıflarından hangisine ait olduğu tahmin edildiği için diziden tek bir çıktı elde edilir. Yani diziden-bire (sequence-to-one) bir öğrenme söz konusudur.

RNN'lerin geriye yönelik adımları hatırlama yaklaşımında kaybolan (vanishing) gradyan ve patlayan (exploding) gradyan gibi iki önemli sorunla karşılaşılır. Bu problemlerin çözümüne yönelik LSTM'lerin uzun-kısa hafıza etkinliği sağlayan eğitilebilir kapıları önerilmiştir. Eğitim boyunca, bir sonraki durumun tahmininde hesaplanan genel RNN model yapısıyla birlikte giriş, çıkış, öğrenme, unutma gibi farklı fonksiyonlara sahip kapılar da öğrenme sürecine dâhil edilir. Bu şekilde, modelin geçmişe yönelik hafızasının hangi şartlarda ve hangi oranda sonuca etki edeceğine karar veren bir sistem elde edilir.

Bu çalışmada LSTM ve BiLSTM modellerinin başarısı ayrı ayrı incelenmiştir. Önerilen standart LSTM modelinin katmanları ve önemli parametrelerine ait detayları Şekil 3.12'de verilmiştir.

```
_model_name == "LSTM":
_model = Sequential()
_model.add(embedding_layer)
_model.add(SpatialDropout1D(0.2))
_model.add(LSTM(128, dropout=0.3, recurrent_dropout=0.3))
_model.add(Dense(num_of_classes, activation='softmax'))
```

Şekil 3.12. LSTM modeli özellikleri

Bi-LSTM, standart bir LSTM'nin çift yönlü versiyonudur. Diğer RNN varyantlarının da çift yönlü versiyonları bulunmaktadır. Çift yönlü RNN arkasındaki fikir; ileriye doğru tekrarlayan

katmanlara ek olarak geri yönlü hesaplama yapan katmanların kullanılmasıdır. Diziler baştan sonra ve sondan başa ayrı ayrı işleme alınır. Tek yönlü bir RNN'de $x_1, x_2, x_3 \dots x_N$ girişleri tekrarlayan bir bağıntıyla $y_1, y_2, y_3 \dots y_N$ çıkışlarını verir. Çift yönlü RNN'de buna ek olarak geri yönde $x_N,\ x_{N-1}, x_{N-2} \dots x_1$ girişleri ile $\widehat{y_1}, \widehat{y_2}, \widehat{y_3} \dots \widehat{y_N}$ çıkışları elde edilir. Bu iki yönlü çıkışlar bir araya getirilerek ağın nihai çıktısı elde edilir.

3.4.4. CNN ve LSTM Modellerinin Bir Arada Kullanılması

Literatürde CNN ve LSTM modellerinin farklı mimariler içerisinde bir arada kullanılmasıyla çeşitli problemlerde başarılı sonuçlar elde edildiği görülmektedir [89–91]. Bu çalışmada ele alınan probleme özgü olarak iki farklı hibrit model önerilmiştir. BiLSTM-CNN ve CNN-BiLSTM olarak isimlendirilen bu hibrit modellerin birbirlerine göre başarısının yanı sıra yalın LSTM ve CNN modellerine göre gösterdiği performans avantajları kapsamlı deneylerle ele alınmıştır. CNN-BiLSTM modelinin detaylı içyapısı Şekil 3.13'te sunulmuştur.

```
_model_name == "CNNBiLSTM":
_model = Sequential()
_model.add(embedding_layer)
_model.add(Dropout(0.2))
_model.add(Conv1D(filters=64, kernel_size=3, activation='relu'))
_model.add(MaxPooling1D(pool_size=2))
_model.add(Bidirectional(
    LSTM(128, return_sequences=True, dropout=0.3, recurrent_dropout=0.3)))
_model.add(Flatten())
_model.add(Dense(128, activation='softmax'))
_model.add(Dropout(0.1))
_model.add(Dense(num_of_classes, activation='softmax'))
```

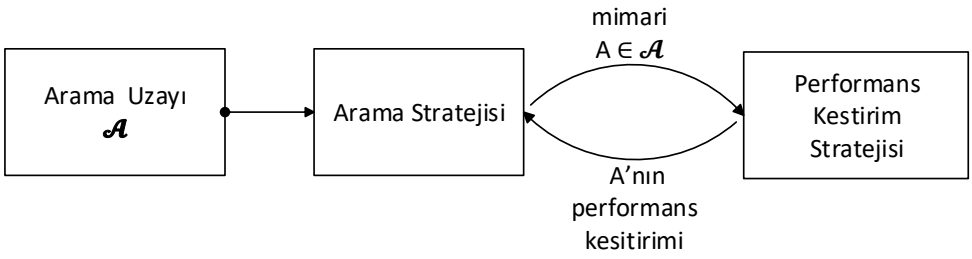
Şekil 3.13. CNNBiLSTM modeli özellikleri

Şekil 3.13'te de görüleceği üzere ilk aşamada kelime gömme katmanına tabi tutulan kullanıcı yorumlarında oluşan vektörler evrişim ve ortaklama katmanlarından sonra iki yönlü LSTM'den geçmektedir. Son olarak LSTM çıkışları yoğun bir katmana verilmek üzere sıralı bir şekilde ayarlanarak (flatten) softmax sınıflandırıcıyla sınıf etiketi tahmin edilmektedir. BiLSTM-CNN modelinde ise benzer bir süreç LSTM katmanlarının CNN ile yer değiştirmesi şeklinde tasarlanmıştır. Bu modellerin diğer yalın alt modellerle sağlıklı bir şekilde karşılaştırabilmesi amacıyla parametreleri de diğer modellere benzer seçilmiştir. Örneğin LSTM katmanı için nöron sayısı 128 olarak belirlenmiştir.

3.5. Hiper Parametre Araması

Derin öğrenme yönteminde diğer makine öğrenmesi yöntemlerinde olduğu gibi belirli sayıda parametre ile problemin eğitim verilerini maksimum düzeyde doğru tahmin eden bir modelleme yapılır. Model eğitimi olarak adlandırılan bu süreçte kurgulanan derin öğrenme mimarisine göre nöronların ağırlıkları öğrenilir. Model ağırlıklarının yüklenme şekli, hangi aktivasyon fonksiyonlarının kullanılacağı gibi modelin yapısıyla ilgili dâhili etmenler başarılı bir model üretme sürecinde kritik rol oynar. Hiper parametre olarak adlandırılan bu etmenler öğrenme katsayısı, iterasyon sayısı gibi harici hiper parametrelerle birlikte doğru olarak seçildiği takdirde istenen başarı elde edilebilir. Fakat parametrelerin doğru seçilmesine rehberlik edecek genel kabul görmüş formüller bulunmamaktadır.

Hiper parametre seçimi, çözülmüş bir problem değildir. Ancak, parametrelerin alabileceği değerler üzerinden çeşitli manuel deneme yanılma yöntemleri ve algoritmik arama stratejileri önerilmiştir. Elsken ve arkadaşları [92], derin öğrenme tabanlı çözümler için model arama problemine yönelik önerilen yaklaşımları arama uzayı, arama stratejisi, performans ölçüm stratejisi ana hatlarıyla kategorize etmiştir. Şekil 3.14'te verilen model yapısının belirlenmesi yaklaşımlarını özetleyen bu gösterim literatürdeki hiper parametre aramaları için bir özet olarak değerlendirilebilir. Örneğin model eğitimini gerçekleştiren araştırmacının kendi bilgi ve tecrübelerinden yola çıkarak katman sayısının uygun olacağı sayısal değerleri belirlemesi sezgisel bir stratejidir. Bu değerler, belirlenen performans ölçüm stratejisiyle karşılaştırılarak aralarından en uygun olan değer seçilir. Bu yaklaşımın en önemli dezavantajı, arama uzayının araştırmacının önsezileriyle (intuition) sınırlı olmasıdır.



Şekil 3.14. YSA tabanlı mimari arama yöntemleri [92]

Literatürdeki en yaygın hiper parametre arama yöntemlerinden biri ızgara aramasıdır. Izgara arama (grid search), hiper parametre uzayının sınırlı bir alt kümesi için parametreleri çaprazlayarak elde edilen kombinasyonların tahlilini yapar [93]. Bu konudaki bir diğer önemli yöntem Benjio ve arkadaşları tarafından önerilen rasgele aramadır (random search) [94]. Grid arama yöntemindeki gibi muhtemel bütün parametre kombinasyonlarının denenmesi yerine

rasgele seçilen alt kümeler üzerinden hata değerleri hesaplanır. Derin öğrenme yöntemlerinin temel hiper parameteleri şunlardır:

Gizli Katman Sayısı

Genellikle kullanılan katman sayısı arttıkça daha derin bir öğrenme gerçekleştirileceği düşünülebilir. Fakat belirli bir sayının üzerinde katman eklemek verilerin ezberlenmesine ve test verileri üzerindeki başarının azalmasına neden olmaktadır. Derin öğrenme modellerinin hiyerarşik olarak otomatik özellik çıkartımı gerçekleştirdiği düşüncesiyle, problem ve veri türüne bağlı olarak çeşitli özellik seviyelerinin öğrenilmesi sezgisel olarak hedef alınabilir. Bu bağlamda, örneğin nesne tanıma probleminde; kenar, şekil, renk bilgilerini öğrenmek üzere katmanlar atfedilebilir. Doğal dil işleme problemlerinde ise model tarafından öğrenilmesi istenen dilin yapısına bağlı özellikler dikkate alınarak gerekli katman sayısı üzerinde tahmin yürütülebilir.

Gizli katman sayısı, katmanda bulunan **nöron sayısı** parametresi ile birbirine girift bir şekilde bağlıdır. Nöron sayısı her katman için farklı seçilebilir.

Aktivasyon Fonksiyonu

Transfer fonksiyonu olarak da bilinen bu fonksiyon ile ağırlıklı giriş değerleri toplamının bir sonraki hücreye nasıl aktarılacağı tanımlanır. Aktivasyon fonksiyonu seçimi modelin öğrenme yeteneğinde kritik bir rol oynar. Aktivasyon fonksiyonu katmanlar arasında farklılık gösterebilir [95]. Aktivasyon fonksiyonu çeşitleriyle ilgili bilgiler Bölüm 2.3.3'te verilmiştir.

Öğrenme Oranı (Learning Rate - LR):

Genellikle 0 ile 1 arasında 0.1, 0.01, 0.003 gibi seçilen bir değer olup, modelin mevcut parametrelerle gerçekleştirdiği tahmin sonucunda elde edilen hata miktarının modelde ne büyüklükte bir değişim meydana getireceğini kontrol eden bir hiper parametredir. "$Yeni\_Ağırlık = Mevcut\_Ağırlık$ — Öğ$renme_Oranı\ X\ Gradyan$" şekilde formüle edilebilir. Düşük bir öğrenme oranı, ağın daha hızlı eğitilmesini sağlar, ancak minimum kayıp işlevinin eksik olmasına neden olabilir.

Seyreltme (Dropout)

Aşırı öğrenme (overfitting) probleminin önüne geçmek amacıyla model eğitimi sırasında bazı nöronların rastgele seçilerek devre dışı bırakılması yöntemidir. Seyreltme, bir çeşit regülarizasyon görevi görerek bütün nöronların aynı anda öğrenmesi yerine farklı kombinasyonlarda devrede olmalarıyla, veri seti üzerinde daha iyi bir genelleme sağlayan verimli ve popüler bir tekniktir. İlk kez, derin öğrenme alanının öncülerinden olan Geoffrey Hinton ve arkadaşları tarafından önerilmiştir [96].

Ağırlıkların Başlangıç Değerlerinin Atanması (Weights Initialization)

Modelin ilk örnekle karşılaşmadan önceki başlangıç durumunda ağırlıkların hangi değerlere sahip olacağına karar verilir. Bütün değerler başlangıçta sıfır yapılabilir, belirli aralıktaki rastgele sayılar verilebilir veya özel bir değer üretme yöntemi kullanılabilir. Yapay sinir ağlarında, kullanılan aktivasyon fonksiyonuna uygun ağırlık başlangıç değerlerinin atanması modelin performansında önemli bir rol oynamaktadır [97].

İterasyon, Epoch, Batch, Mini-batch

Eğitim verisinin tamamının sinir ağından bir kez geçirilmesi ve geri yayılım ile model ağırlıklılarının güncellenmesinin tamamlanmasına **epoch** (dönem) denir.

Eğitim setlerinde genellikle bir epoch için çok sayıda örnek bulunmaktadır. Bundan dolayı bütün veri alt parçalara ayrılarak hesaplamaya dâhil edilir. **Mini-batch** adı verilen her bir alt parça üzerinden gerçekleştirilen adımlara **iterasyon** denir. Örneğin 1024 adet örneğe sahip bir eğitim setinin kullanıldığı eğitimde mini-batch boyutu 64 olarak seçildiği takdirde 1 epoch'un tamamlanması için 16 iterasyon gerçekleştirilir. Mini-batch boyutu 1024 olarak, yani veri seti büyüklüğünde seçildiğinde ise bu dilime **batch** denir. Mini-batch boyutunun seçimi probleme özgüdür. Literatürde 32, 64, 128 ve 256 değerlerinin tercih edilmesine sıklıkla rastlanmaktadır.

Bu değerlerin seçimi, gradyan iniş algoritması açısından önemli bir etkiye sahiptir. Sinir ağının tek bir örnek için (mini batch = 1) hata değerini hesaplayarak, gradyan alıp geri yayılımı gerçekleştirmesi sonucunda ağırlıkların güncellenmesi **Stokastik Gradyan İniş** yaklaşımının kullanılması demektir. Bu işlemin 1'den büyük ve veri setinin tamamından küçük bir değer (mini batch) için gerçekleştirilmesi durumunda örnekler arasında gradyan değerinin ortalaması alınır, yani **Mini-Batch Gradyan İniş** kullanılır. **Batch Gradyan İniş**'te ise veri setinin tamamı işleme alınır ve gradyanın bütün veriler için ortalaması kullanılır.

Optimizasyon Yöntemleri

Gradyan iniş, makine öğrenmesinde ve daha sonra yapay sinir ağlarındaki gelişmelerle birlikte derin öğrenme yöntemlerinde kullanılan temel bir optimizasyon yöntemidir. Derin öğrenme modelinin mevcut parametreler ile eğitim verisi üzerinde göstermiş olduğu hata miktarını, bir diğer ismiyle maliyeti minimize etmek için gradyan iniş yöntemleri kullanılır. Maliyet hesabı, gradyan alınması ve parametrelerin güncellenmesi döngüsünde gerçekleşen optimizasyon sürecinde her bir dönemde kullanılan örnek sayısı oranına göre mini-batch, stokastik gibi farklı gradyan iniş yöntemlerinin olduğu önceki tanımda açıklanmıştır.

Derin öğrenme yöntemlerinin son derece hızlı bir şekilde gelişmesiyle birlikte gradyan iniş tabanlı optimizasyon yöntemlerinde dikkat çekici yenilikler ortaya çıkmıştır. Standart bir gradyan iniş yaklaşımında sabit olan öğrenme oranı değerinin adaptif olarak kendisini güncellediği farklı

algoritmalar sayesinde çeşitli problemlerde performans artışları sağlanmıştır [98]. Root Mean Square Propagation (RMSProp) [99] ve Adaptive Moment Estimation (Adam) [100] optimizasyon yöntemleri oldukça sık tercih edilmektedir. RMSProp, derin öğrenme modellerinde parametreler katmanlarla birbirlerinden seviyeler hâlinde ayrıldığı için gradyanların ilk katmanlara yayılımına kadar geçen adımlar sonrasında sıfır değerine yaklaşması olarak bilinen kaybolan gradyan (vanishing gradient) problemine yönelik geliştirilmiştir. Stokastik bir mini-batch öğrenme tekniği olan RMSProp yöntemi bu çalışmadaki bütün derin öğrenme modellerinde tercih edilen optimizasyon algoritmasıdır.

3.6. Yöntem ve Performans Ölçütleri

Sınıf dengesizliği problemi destek vektör makineleri, karar ağaçları ve sinir ağları gibi yaygın sınıflandırıcıların tamamının performansını olumsuz etkilemektedir. Sınıf dengesizliği ile başa çıkmak için geliştirilen yöntemler iki ana gruba ayrılabilir; dâhili yöntemler, spesifik olarak düzensiz veri kümeleriyle başa çıkmak üzere tasarlanmış yeni algoritmaların geliştirilmesiyle ilgilenirken, harici yöntemler, veri setinin dengesizliğini hafifletmek üzere sınıflandırıcının eğitimi için kullanılacak veri setiyle ilgilenir [101]. Bu çalışmada harici bir yöntem olan aşırı örnekleme tekniğinin farklı derin öğrenme yöntemleriyle uygulanması kapsamlı olarak ele alınmıştır. Bu tekniğin farklı örnekleme oranlarıyla uygulanmasının modeller üzerindeki başarısı farklı performans ölçütleriyle ortaya konulmuştur.

Farklı mimarilere sahip 6 derin öğrenme modelinin dengesiz ve farklı kaynaklardan elde edilmiş büyük miktarda veri üzerinde duygu sınıflandırma başarısı detaylı olarak incelenmiştir. Modellerin aşırı örnekleme (over sampling) uygulanması ile gösterdiği performans değişimleri; gerçek pozitif, yanlış pozitif, gerçek negatif, yanlış negatif, kesinlik, hassasiyet, doğruluk, seçicilik, dengeli doğruluk ve F-Ölçümü değerleriyle kapsamlı olarak analiz edilmiştir. Bu değerlerin hesaplanmasının temelini oluşturan karmaşıklık matrisi açıklamaları Tablo 3.3'te yer almaktadır.

3.6.1. Performans Ölçütlerinin Genel Tanımları

Bir sınıflandırıcının, gerçek değerleri bilinen bir dizi test verisi üzerindeki performansının açıklanmasında genellikle karmaşıklık matrisinden yararlanılır. Dengesiz veri setleri için bu değerlerin hesaplanmasında bazı önemli detaylara dikkat edilmelidir. Bu bölümde performans ölçütlerinin genel tanımları verilmiştir. Karmaşıklık matrisi açıklamaları Tablo 3.3'te verilmiştir.

Gerçek pozitif (True Positive – TP): Pozitif tahmin edilen örneğin gerçekte pozitif olduğu ***doğru*** bir tahmindir.

Yanlış pozitif (False Positive – FP): Pozitif tahmin edilen örneğin gerçekte negatif olduğu ***yanlış*** bir tahmindir.

Gerçek negatif (True Negative – TN): Negatif tahmin edilen örneğin gerçekte negatif olduğu ***doğru*** bir tahmindir.

Yanlış negatif (False Negative – FN): Negatif tahmin edilen örneğin gerçekte pozitif olduğu ***yanlış*** bir tahmindir.

Tablo 3.3. Karmaşıklık matrisi açıklamaları

	Tahmin edilen **0**	Tahmin edilen **1**
Gerçek değeri **0**	*Gerçek negatif* **TN**	*Yanlış pozitif* **FP**
Gerçek değeri **1**	*Yanlış negatif* **FN**	*Gerçek pozitif* **TP**

Çalışma sonuçlarının değerlendirildiği diğer performans ölçütleri ise TP, TN, FP, FN değerleri üzerinden hesaplanmıştır.

Kesinlik (Precision): Pozitif tahmin edilen örneklerden kaç tanesinin pozitif olduğunu gösterir.

$$Precision = \frac{TP}{TP + FP} \tag{3.1}$$

Hassasiyet (Recall): Pozitif örneklerin doğru tahmin edilme oranını gösterir. Bu değer aynı zamanda Gerçek Pozitif Oran (True Positive Rate – TPR) olarak da bilinir.

$$Recall = \frac{TP}{TP + FN} \tag{3.2}$$

Seçicilik (Gerçek Negatif Oran – TNR): Negatif örneklerin doğru tahmin edilme oranını gösterir.

$$TNR = \frac{TN}{TN + FP} \tag{3.3}$$

Dengeli Doğruluk (Balanced Accuracy – B_ACC): Gerçek negatif oranı ile gerçek pozitif oranının ortalamasıdır.

$$B\_ACC = \frac{TNR + TPR}{2} \tag{3.4}$$

Doğruluk (Accuracy – ACC): Tahminlerin doğruluk oranıdır. Genel başarı ölçütü olarak kullanılır.

$$ACC = \frac{TP + TN}{TP + FP + TN + FN} \quad (3.5)$$

F-Ölçümü (F1 Skoru): Hassasiyet ve kesinlik değerlerinin harmonik ortalamasıdır.

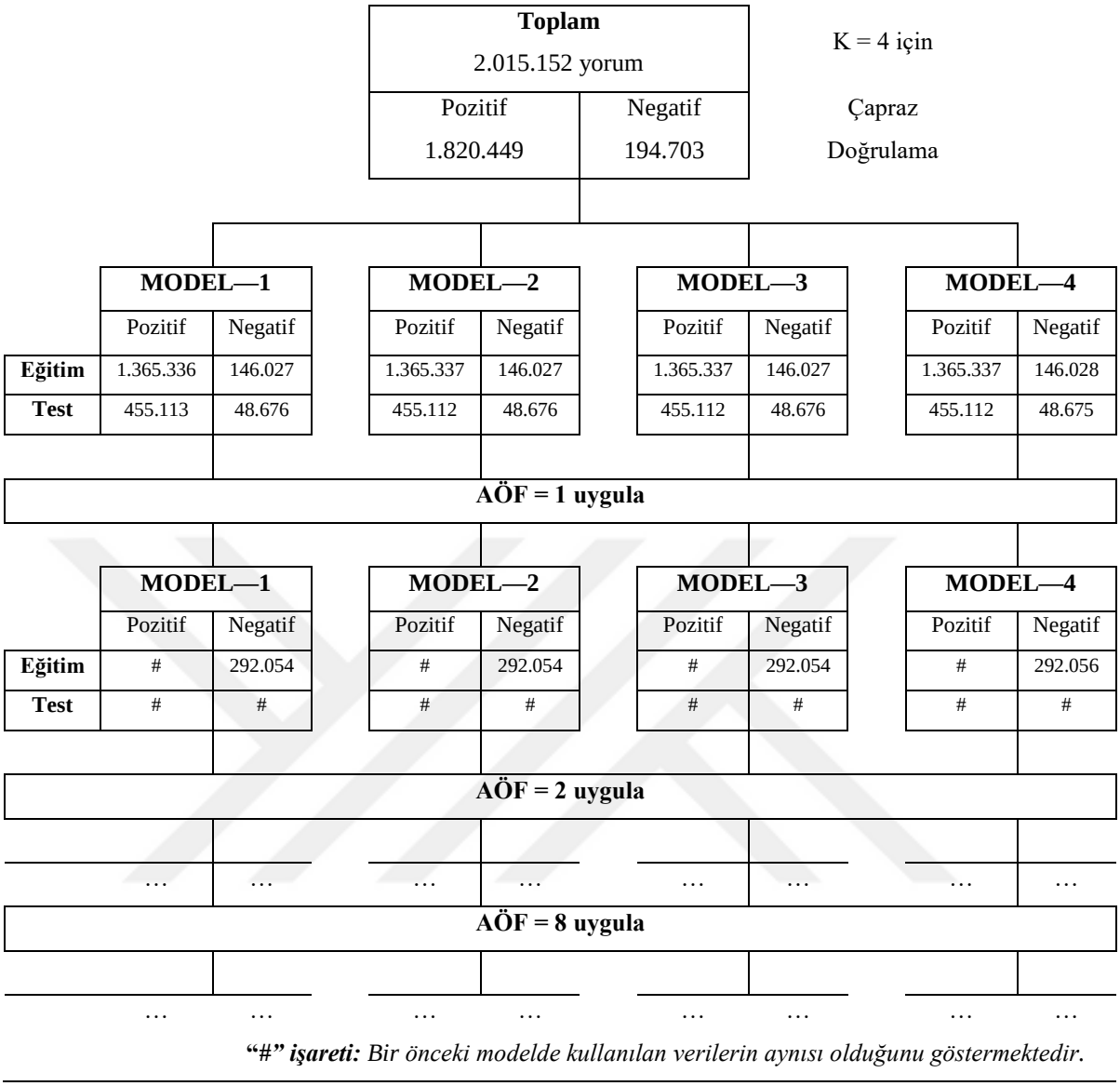
$$F1 = \frac{2\ X\ Precision\ X\ Recall}{Precision + Recall} \quad (3.6)$$

3.6.2. Dengesiz Veri Seti Problemi ve Aşırı Örnekleme Uygulaması

Veri seti açıklamalarında belirtildiği üzere negatif örnek (negatif yorumlar) ve pozitif örnek sayıları arasında büyük bir dengesizlik söz konusudur. Yorumların sadece bir kaynaktan değil farklı alanlarla ilgili birden fazla kaynaktan elde edildiği dikkate alınırsa, bu durumun duygu analizi problemi için yaygın bir koşul olduğu düşünülebilir. Literatürdeki derin öğrenme tabanlı duygu analizi çalışmalarında bu ölçüde bir veri dengesizliğinin ele alındığı, Türkçe diline ait, farklı web sitelerinden elde edilen yorumlarla gerçekleştirilen bir çalışma bildiğimiz kadarıyla mevcut değildir. Genellikle bir veya iki kaynaktan elde edilen, on binlerle ifade edilecek büyüklüklerde, görece dengeli veri setlerinin kullanıldığı görülmektedir [77].

Uygulamalar iki aşamada gerçekleştirilmiştir. Öncelikle, veri seti dengesizliğine yönelik herhangi bir çözüm önermeden, farklı derin öğrenme modellerinin duygu sınıflandırmadaki performansları araştırılmıştır. Daha sonra, bu modellerin farklı oranlarda aşırı örneklemeye karşı gösterdiği performans değişimleri gözlenmiştir. Bütün deneyler K-Sayısı Kadar Çapraz Doğrulama (K-Fold Cross Validation) sonuçları üzerinden değerlendirilmiştir. Şekil 3.15'te gerçekleştirilen deneysel çalışmaların detayları sunulmuştur. Şekilde sadece bir derin öğrenme mimarisinin incelenmesi açıklanmıştır. Aynı süreç, incelenen 6 farklı derin öğrenme modeli için tekrarlanmıştır. Her aşamada kullanılan eğitim ve test örnek sayıları, pozitif ve negatif sınıflara göre verilmiştir.

K-Fold Cross Validation uygulandığında veri setinin tamamı K adet parçaya bölünür. Örneğin bu değer 4 olarak seçilirse, eşit büyüklükteki 4 alt parça (fold) elde edilir. Bu parçaların üçünün (K-1) eğitim ve birinin test verisi şeklinde kullanılmasının kombinasyonlarıyla eğitilen 4 farklı model elde edilir. Böylelikle, farklı dağılımlara sahip eğitim ve test verileri üzerinde modelin genel bir performans değerlendirmesi yapılır.



Şekil 3.15. Aşırı örnekleme ve K-Fold Çapraz Doğrulama deneyleri

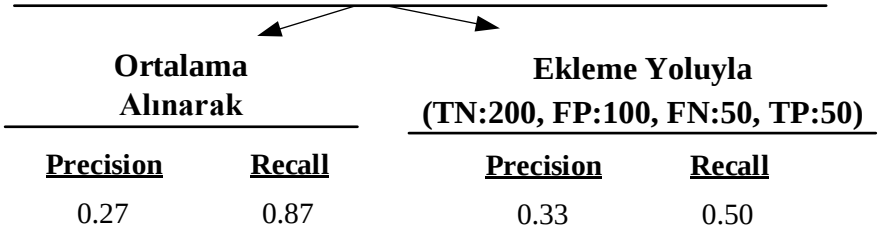
Veri setinin %80'i eğitim, %20'si test şeklinde ayrılması gibi sabit bir dağılımda, önerilen modeller arasındaki performans karşılaştırması çeşitli tesadüfi faktörlere bağlı olarak gerçeği yansıtmayabilir. Çözülmek istenen asıl problem, veri setinin negatif örnekler açısından pozitif örneklere oranla çok zayıf olmasıdır. Bu durum, model başarılarının değerlendirilmesinde iki önemli noktaya dikkat edilmesi gerektiğini göstermektedir:

1. Sadece doğruluk değeri (accuracy) üzerinden gerçekleştirilecek değerlendirmeler sağlıklı değildir. 2 adet negatif, 98 adet pozitif örneğe sahip bir test kümesi üzerinde yapılan tahminlerin her koşulda ezbere olarak pozitif şeklinde yapılması %98 başarı demektir. Fakat bu sonuç gerçekçi değildir. Daha sağlıklı bir değerlendirmenin yapılması için pozitif ve negatif sınıfların hangi oranlarda doğru bilindiği ayrı ayrı ele alınarak tahminlerdeki kesinlik, hassasiyet ve seçicilik gibi yönler dikkate

alınmalıdır. Bu nedenle çalışmada genel başarı değerinin yanı sıra Precision, Recall, TNR, B_ACC ve F1 değerleri hesaplanmıştır.

2. Precision, recall gibi değerler normalde sadece bir model için hesaplanan skorlardır. Bu çalışmada, farklı veri dağılımları noktasında, modeller arasında mümkün olduğunca genel bir karşılaştırma yapabilmek için K sayısı kadar çapraz doğrulama değerlendirmesi uygulandığından dolayı her bir model yapısı için (CNN, LSTM, vb.) K adet model eğitilmiştir. Örneğin 4 farklı CNN modeline ait başarı değerleri bulunmaktadır. Bu değerlerin bir arada nasıl ele alınacağı belirlenmiştir.
3. Bir modelin başarısı, 4 farklı versiyonu üzerinden skorların ortalaması şeklinde belirlenebilir. Ortalama alma yaklaşımının ne kadar sağlıklı olduğu konusunda fikir vermesi amacıyla Şekil 3.16'da örnek bir hesaplama gösterilmiştir. Yapılan hesaplama bu yöntemin gerçek başarıyı yansıtmadığını ortaya çıkarmıştır. Örneğin, 4 model toplamda 100 pozitif örnekten sadece 50 tanesini doğru tahmin etmiştir. Fakat ortalama alındığında ilk 3 model kendi test kümelerinde yer alan 1'er pozitif örneği doğru tahmin ettiğinden dolayı pozitif örnekleri doğru tahmin etme oranını (Recall) örnek sayısına göre orantısız olarak %87'ye yükseltmektedir [102].

Fold	TN	FP	FN	TP	Precision	Recall
Fold1	79	20	0	1	0,05	1
Fold2	69	30	0	1	0,03	1
Fold3	49	50	0	1	0,02	1
Fold4	3	0	50	47	1	0,49



Şekil 3.16. K-Fold için model performans tespiti analizi

3.7. Uygulamalar

Derin öğrenme yöntemlerinden MLP, CNN, LSTM, BiLSTM gibi temel mimarilerle birlikte aralarında en yüksek başarıyı göstermesi dikkate alınarak BiLSTM modelini CNN ile iki farklı şekilde bir araya getirerek oluşturduğumuz hibrit BiLSTM-CNN ve CNN-BiLSTM modelleri tercih edilmiştir. Modellerin iç yapısı ve seçilen temel hiper parametreler Bölüm 3.4'te derin öğrenme modellerinin anlatımı ile birlikte verilmiştir.

Hesaplama ortamı olarak 2 adet GPU-sunucu kullanılmıştır. Her bir makinede 8 adet **11GB** hafızalı **Nvidia GeForce GTX 1080 Ti** ekran kartı bulunmaktadır. Makineler, **48-core Intel Xeon CPU E5** işlemci ve **256GB RAM** hafızaya sahiptir. Model eğitimleri TensorFlow ve Keras kütüphaneleri kullanılarak ekran kartları üzerinde gerçekleştirilmiştir.

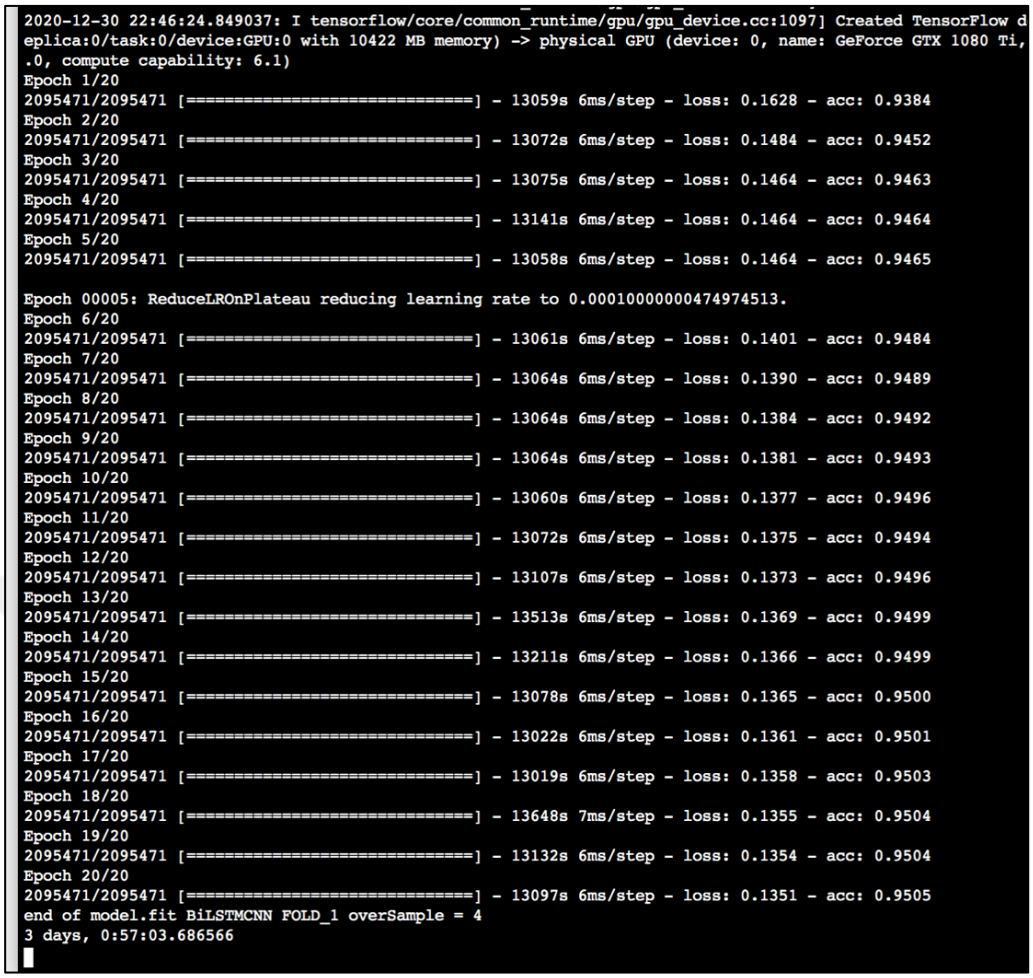
Deney adımlarında 4-Sayısı Kadar Çapraz Doğrulama uygulandığı Şekil 3.15'te belirtilmiştir. Sırayla farklı parçaların (fold) test verisi olarak ayrıldığı eğitimler, BiLSTM-CNN gibi parametre sayısı çok fazla olan modeller için 20 dönem (epoch) yaklaşık 4 gün sürebilmektedir. Deneylerin, zaman ve hesaplama kaynağı bakımından oldukça maliyetli olması da göz önünde bulundurularak K-sayısı Kadar Çapraz Doğrulama için K değeri 4 olarak belirlenmiştir.

Çok detaylı olmamakla birlikte hiper parametre araştırması yapılmıştır ve ana deneylerde daha iyi başarı gösteren değerler seçilmiştir. Farklı parametrelerin, modellerin kendi içerisindeki başarıları üzerinde gösterdiği performans karşılaştırmalarına değinilmemiştir. Bunun yerine, modellerin bilinen standart yapılarıyla, çok fazla parametreyle karmaşıklaştırılmadan birbirleri arasında tarafsız bir karşılaştırma yapılmıştır. Örneğin, CNN modelinin en başarılı versiyonunu araştırmak yerine, uygulanan aşırı örnekleme tekniklerinin BiLSTM, CNN-BiLSTM gibi farklı modellerden hangisinde daha iyi performans sağladığı gösterilmiştir. Veri sayısının büyük olması ve başarı değerlendirmesinde K-sayısı Kadar Çapraz Doğrulama gibi maliyetli yöntem kullanılması dolayısıyla tüm olasılıklar için deneme yanılma yoluna gidilmesinin doğru bir yöntem olmayacağı düşünülmüştür. Problem için en iyi veri seti yapısı ve derin öğrenme yöntemi ortaya konularak literatüre katkı sağlanmıştır.

Kategorik çapraz düzensizlik (categorical cross-entropy), bu çalışmadaki bütün modellerde tercih edilen hata fonksiyonudur. Öğrenme oranı (LR) başlangıç değeri de bütün modeller için aynı tutulmuştur. Eğitim içerisinde seçilen metriklerin bir ilerleme kaydetmediği durumda LR değerinin otomatik olarak düşürülmesi sağlanmıştır. Bunun için Keras kütüphanesinde tanımlı ***ReduceLROnPlateau*** konfigürasyonundan ($monitor = 'loss',\ patience = 2, epsilon = 1e-4,\ factor = 0.1,\ mode = 'min'$) yararlanılmıştır. Bu yapılandırma model eğitimini şu şekilde yönlendirmektedir:

1. Modelin hata oranı (**loss**) gözlenir.
2. 2 epoch üst üste loss değerinde bir azalma olup olmadığına bakılır (**patience**).
3. Bir değişiklik olduğunun kabul edilmesi için hata oranının en az 0.0001 miktarında değişmesi gerekir (**epsilon**).
4. Tanımlanan şartların (1-2-3) gerçekleştiği durumlarda yeni LR değerini azalt (**mode**): $Yeni\_LR\ = LR * \boldsymbol{factor}$

BiLSTM-CNN modelinin eğitimi sırasında çalışma ortamından alınan ekran görüntüsü ile uygulama ortamı için açıklanan ReduceLROnPlateau kullanımı gibi detaylar ve hesaplama ortamı özellikleri Şekil 3.17'de görülmektedir.



Şekil 3.17. Eğitim ortamı ve ReduceLROnPlateau etkisi

Verilerin K adet parçaya bölünmesi rasgele yapılmıştır. Fakat bu rasgele dağılım deneyler arasında sabit tutulmuştur. Bunun için Python *seed()* fonksiyonu kullanılmıştır. Verilerin karıştırılması için de yine aynı sözde rasgele sayı üreteci mekanizmasından yararlanılmıştır. Aşırı örnekleme faktörü (AÖF) adında bir harici parametre belirlenmiştir. Bu parametre miktarı az olan negatif verilerin katlanma sayısını tanımlamaktadır. Negatif verilerin tekrarlanarak veri setine yeniden eklenmesi şeklinde uygulanan aşırı örnekleme süreci aşağıdaki 3 adımda açıklanmıştır:

1. Verileri karıştır (random.seed)
2. K adet parçaya böl
3. K. parçanın test verisi olduğu K. model için
    a. Eğitim verisinde yer alan negatif örnekleri bul
    b. AÖF sayısı kadar bu örnekleri tekrarla ve eğitim verisiyle birleştir
    c. Eğitim verisini yeniden rasgele karıştır
    d. Model eğitimini başlat

Aşırı örnekleme yaklaşımının uygulandığı modeller de dâhil olmak üzere her K-fold dağılımı için bütün deneylerde test verisi aynı tutulmuştur. Aynı şekilde eğitim verilerinde yer alan pozitif örnekler de sabit olup sadece karıştırılmıştır (shuffle). Veri setindeki pozitif örnekler, negatif örneklerin yaklaşık 9 katıdır ($1{,}820{,}449 / 194{,}703$). Negatif örnek sayısı 8 kez tekrarlanıp veri setine eklendiğinde pozitif ve negatif örnek sayıları arasında bir denge kurulmaktadır. Bundan dolayı gerçekleştirilen deneylerde AÖF 1, 2, 4 ve 8 değerleri üzerinden incelenmiştir.

3.8. Sonuçlar ve Değerlendirme

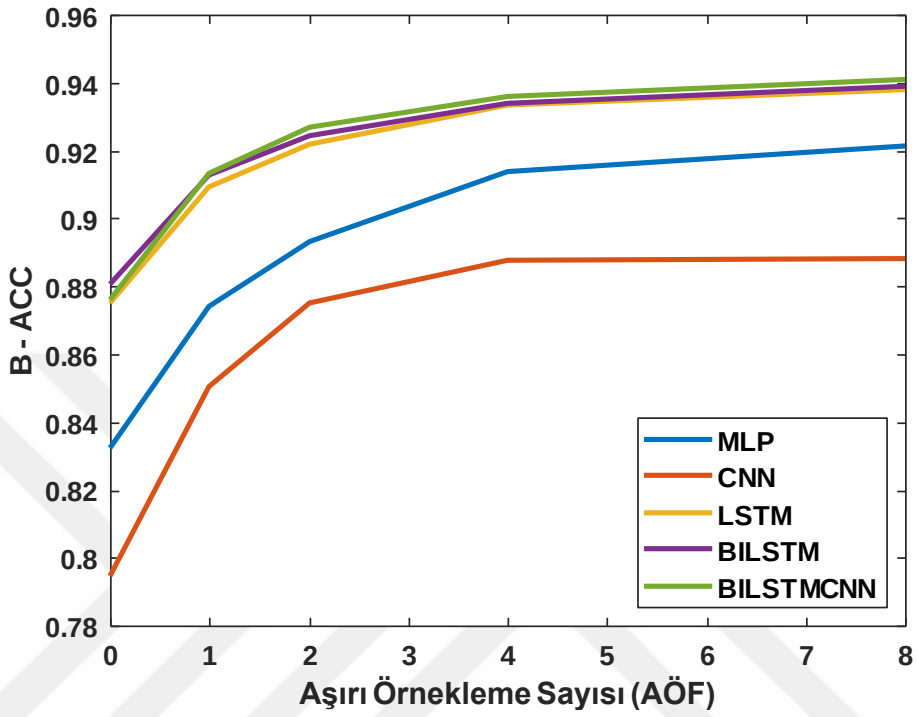
Şekil 3.15'te gösterilen deney adımlarının bütün modeller için gerçekleştirilmesi; 6 farklı modelin farklı miktarlarda aşırı örnekleme uygulanan dört durum ve hiç uygulanmadığı bir durum için olmak üzere 30 farklı şekilde eğitimi anlamına gelmektedir. Bu 30 eğitimin her birinde 4-sayısı Kadar Çapraz Doğrulama uygulanması nedeniyle toplam 120 deney söz konusudur. CNN-BiLSTM modelinin henüz hiçbir aşırı örnekleme stratejisi uygulanmadan gösterdiği performans diğer modellerin başarısının altında olduğundan dolayı bu modelin AÖF sonuçları elde edilmemiştir. Bu modelin, en az CNN ve BiLSTM modellerinin ayrı ayrı gösterdiği başarının üzerinde bir performans göstermesi gerektiği yönünde bir değerlendirme yapılmıştır.

Tablo 3.4. Modellerin aşırı örnekleme öncesi performansları

Model	Precision	Recall	TNR	F1	ACC	B_ACC
MLP	.9661	.9894	.6757	.9776	.9591	.8325
CNN	.9586	.9882	.6013	.9732	.9508	.7947
LSTM	.9749	.9881	.7622	.9815	.9663	.8751
BiLSTM	.9761	.9873	.7740	.9667	.9816	.8806
BiLSTMCNN	.9750	.9893	.7628	.9821	.9674	.8760
CNNBiLSTM	.9548	.9888	.5628	.9715	.9477	.7758

Tablo 3.4'de, dengesiz veri problemine yönelik herhangi bir çalışma yapılmadan elde edilen genel başarı değerleri verilmiştir. BiLSTM-CNN modelinin F1 skoru yönünden en iyi, B_ACC değeri yönünden en iyi ikinci başarıyı gösteren derin öğrenme yöntemi olduğu görülmektedir. BiLSTM ise TNR ve B_ACC başarısı en yüksek olan modeldir. TNR değeri %56.28 olan CNN-BiLSTM yönteminin F1 skoru bakımından %97.15 sonuç vermesi, F1 skorunun bu problem için iyi bir performans değerlendirme kriteri olmadığını göstermektedir. Bu model negatif örnekleri %43.72 ihtimalle hatalı tahmin etmesine rağmen yüksek F1 skoruna sahiptir. Veri setinin büyük bir bölümü pozitif örneklerden oluştuğu için modelin en kritik yeteneği çok az miktarda olan

negatif örnekleri de doğru tahmin edebiliyor olmasıdır. Dolayısıyla sadece pozitif örneklerin doğru tahmin edilmesinin önemli olduğu problemler dışında F1 yerine B_ACC değerinin daha iyi bir performans ölçütü olduğunu söylemek mümkündür.



Şekil 3.18. AÖF- B_ACC değişimi grafiği

Veri dengesizliğini azaltacak bir yaklaşımın farklı derin öğrenme modelleri üzerinde nasıl bir performans iyileştirmesi sağlayacağının gösterilmesi bu çalışmanın en önemli katkılarından biridir. Burada 2 önemli sorunun cevabı verilmiştir:

- Uygulanan aşırı örnekleme stratejisi model performansını arttırırken modeller arasındaki performans farkının nasıl bir değişim gösterdiği, sonuç olarak hangi modelin en iyi olduğu.
- Aşırı örnekleme yaklaşımı uygulanırken, az miktarda örneğe sahip olan sınıfa ait verilerin hangi oranda çoğaltılması gerektiği.

5 modelin uygulanan AÖF değerine göre B_ACC değerinde değişimi Şekil 3.18'de gösterilmektedir. AÖF değeri 1, 2, 4 ve 8 olarak seçildiğinde en yüksek başarıyı gösteren model BiLSTM-CNN olmuştur. Deneylerin diğer başarı metrikleriyle birlikte detaylı sonuçları Tablo 3.5'te yer almaktadır.

Tablo 3.5. AÖF sayısına göre model başarı değerleri

Model	*AÖF*	Precision	Recall	TNR	F1	ACC	B_ACC
MLP	*1*	.9753	.9811	.7674	.9782	.9604	.8743
	2	.9799	.9731	.8131	.9765	.9576	.8931
	4	.9859	.9556	.8721	.9705	.9476	.9138
	8	.9908	.9229	.9199	.9557	.9226	.9214
CNN	*1*	.9708	.9745	.7262	.9726	.9505	.8503
	2	.9768	.9643	.7863	.9705	.9471	.8753
	4	.9807	.9496	.8257	.9649	.9376	.8876
	8	.9829	.9271	.8491	.9542	.9196	.8881
LSTM	*1*	.9827	.9797	.8383	.9812	.9661	.9090
	2	.9859	.9737	.8697	.9798	.9637	.9217
	4	.9894	.9632	.9034	.9761	.9574	.9333
	8	.9922	.9457	.9304	.9684	.9442	.9381
BiLSTM	*1*	.9836	.9789	.8470	.9812	.9662	.9129
	2	.9866	.9724	.8766	.9795	.9632	.9245
	4	.9895	.9629	.9048	.9761	.9573	.9339
	8	.9921	.9484	.9293	.9698	.9466	.9389
BiLSTMCNN	*1*	.9834	**.9811**	.8453	**.9823**	**.9680**	.9132
	2	.9869	.9750	.8791	.9809	.9657	.9270
	4	.9896	.9670	.9051	.9782	.9610	.9360
	8	**.9924**	.9505	**.9316**	.9710	.9487	**.9411**

4. DERİN ÖĞRENME İLE TWEET SINIFLANDIRMA VE ÖĞRENME AKTARIMI

Sosyal medya platformlarında oldukça büyük miktarda metin, fotoğraf ve video verileri üretilmektedir. Çeşitli ticari ve akademik amaçlarla bu veriler üzerinde sınıflandırma, duygu analizi, ironi tespiti, sahte haber tespiti gibi birçok farklı veri analizi tekniği uygulanarak anlamlı çıkartımlar yapılmaktadır.

Metin sınıflandırma problemlerinde kullanılmak üzere büyük miktarda etiketli sosyal medya (Twitter, Facebook, Instagram, LinkedIn vb.) veri seti oluşturulması zaman ve maddi kaynaklar açısından masraflı bir iştir [103]. Öte yandan otomatik etiketleme yöntemleriyle çevrimiçi haber siteleri, Wikipedia vb. kaynaklardan veri elde edilmesi mümkündür.

Bu çalışmada, Twitter verilerinin sınıflandırılmasında büyük miktarda haber verisinden nasıl yararlanılabileceği fikri güncel derin öğrenme modelleri üzerinden incelenmiştir. Az sayıda etiketli tweet kullanılarak çok miktarda haber verisiyle eğitilmiş modellere alan adaptasyonu (domain adaptation) uygulanmıştır. Önerilen yöntemin başarısının ölçülmesinde öğrenme aktarımı (transfer-learning) ve iyileştirme (fine-tuning) adımlarında kullanılan tweet sayısından çok daha fazla miktarda tweet kullanılmıştır. Bu şekilde, gerçek hayatta karşılaşılabilecek etiketli verinin çok az veya hiç mevcut olmadığı bir tweet sınıflandırma senaryosuna uygun koşullar üzerinden bir deney gerçekleştirilmiştir.

4.1. Problemin Tanımı

Bu bölümde tweet sınıflandırma problemi ele alınmaktadır. En yaygın sosyal medya platformlarından biri olan Twitter platformunda, kullanıcıların oluşturduğu en fazla 280 karakterden oluşan kısa metinler üzerinde önceden tanımlı sınıfların tespiti gerçekleştirilmiştir. Farklı derin öğrenme modelleri üzerinde öğrenme aktarımı teknikleri uygulanarak bu modellerin Twitter dışındaki kaynaklardan elde edilen metinlerden nasıl istifade edeceği araştırılmıştır. Tweet ve haber olmak üzere kendi oluşturduğumuz 2 ana metin veri seti üzerinden gerçekleştirilen uygulamalar ile yöntemlerin alan adaptasyonu başarıları ölçülmüştür. Bu bölümde gerçekleştirilen çalışmalar "Experiments on Fine Tuning Deep Learning Models With News Data For Tweet Classification" isimli bildiri ile sunulmuştur [104].

Twitter verileri, sosyal medya platformlarının doğası gereği URL'ler, semboller, yazım hataları vb. içerdiği için oldukça gürültülüdür. Bu nedenle metin sınıflandırma teknikleri uygulanırken dikkate alınması gereken ilave problemler söz konusu olabilmektedir. Ayrıca, literatürdeki sosyal medya metinlerinin sınıflandırılması ile ilgili çalışmalar incelendiğinde duygu analizi alanı dışında nadir olarak etiketli veri bulunduğu görülmektedir.

Bu çalışmada tweetlerin kültür, ekonomi, politika, spor ve teknoloji konularından birine ait olup olmadığını belirlemek için metin sınıflandırması gerçekleştirilmiştir. Önerdiğimiz yaklaşımla, çok az miktarda etiketlenmiş tweet mevcut olmasına rağmen yüksek doğrulukla sınıflandırma yapıldığı gösterilmiştir. Büyük miktarda haber verisi ile derin öğrenme modelleri eğitilip daha sonra tweet verileri kullanılarak bu modeller üzerinde öğrenme aktarımı (transfer-learning) ve iyileştirme (fine-tuning) adımları uygulanmıştır.

4.2. Derin Öğrenme Literatürü

Başlangıçta bilgisayarlı görü ve görüntü işleme alanlarında önemli gelişmelere yol açarak popülerleşen derin öğrenme alanı, ilerleyen süreçte makine çevirisi, varlık ismi tanıma ve soru cevaplama gibi birçok önemli DDİ probleminin çözümünde yaygın olarak kullanılmıştır. Metin sınıflandırma, birçok farklı uygulama alanına sahip önemli bir DDİ görevidir. Evrişimli Sinir Ağları (CNN) ve Tekrarlayan Sinir Ağlarının (RNN) varyantları metin sınıflandırma problemlerine ilişkin literatürdeki en iyi sonuçları elde eden ve en çok bilinen derin öğrenme modelleridir. Bu çalışmada, Uzun Kısa Dönemli Bellek Ağları (LSTM), çift yönlü LSTM (Bi-LSTM) ve Geçmeli Tekrarlayan Birim (GRU) da RNN modellerinin varyantları olarak ele alınmıştır.

Yakın zamana kadar dizi modelleme görevlerinde RNN varyantlarının kullanılması varsayılan bir tercih olmuştur. Yeni araştırmalar ise bu genel yaklaşımın yeniden değerlendirilmesi gerektiğini göstermektedir. Bai ve arkadaşlarının önerdiği Zamansal Evrişimli Sinir Ağları yaklaşımı [105], CNN'lerin kelime düzeyinde dil modellemesinde güncel olan en iyi sonuçları elde edebileceğini göstermiştir. Kim [106], CNN ile önceden eğitilmiş kelime gömmelerini birlikte kullanarak cümle düzeyinde metin sınıflandırma problemini yüksek doğrulukta sonuçlarla çözebileceğini göstermektedir. Cao ve arkadaşları [107], metinlerin karakter seviyesinde vektör temsillerinin elde edilmesiyle daha iyi sonuçlar elde edilebileceğini, çünkü sözcüklerle ilgili morfolojik bilgileri de bünyesinde barındırması nedeniyle daha iyi performans göstereceğini ve sözlük boyutunun sınırlı olmasından dolayı derlemde yer almayan sözcüklerden kaynaklı performans düşüşlerinin ortadan kalkacağını savunmaktadır.

2006 yılı itibariyle makine öğrenmesi alanının en popüler alt alanlarından biri olan derin öğrenme yöntemlerinin [108] DDİ problemlerine uygulamasında çeşitli kelime gömme yaklaşımlarından yararlanılmaktadır. Etiketsiz büyük bir derlem üzerinden elde edilen yoğun kelime temsillerinin nispeten daha az miktarda etiketli veriye sahip problemlerde kullanılması bir çeşit öğrenme aktarımı uygulamasıdır. Kelime gömme vektör boyutları genellikle 50 ile birkaç yüz arasında değişir. Kelimelerin semantik anlamlarının sabit uzunluklu vektörler ile kodlanmasında Glove [67] ve word2vec [66] yaygın olarak kullanılmaktadır.

4.3. Veri Setleri

Bu çalışmada haber ve tweet olmak üzere 2 farklı veri seti kullanılmıştır. Bu veri setleri **Bigailab-5news-500K** ve **Bigailab-5tweet-35K** olarak isimlendirilmiştir. Veri setlerinde **ekonomi, kültür, teknoloji, politika** ve **spor** olmak üzere 5 farklı kategoriden eşit sayıda haber ve tweet verileri bulunmaktadır.

Haber dokümanları genellikle yayınlandıkları URL içerisinde ait olduğu kategoriye göre dizinlendiği için manuel bir etiketleme gerektirmeden yüksek miktarda elde edilmesi mümkün olan bir veri türüdür. Haber veri setinin oluşturulmasında Java tabanlı RssTurk [109] web kazıyıcı programı yardımıyla çeşitli çevrimiçi haber sitelerinden büyük miktarda haber dokümanı toplanmıştır. Belirlenen 5 farklı kategori için rastgele seçilen 100 bin adet haber bir araya getirilerek Bigailab-5news-500K veri seti elde edilmiştir.

Literatür incelendiğinde duygu sınıflandırma haricinde metin sınıflandırma çalışmalarında kullanılabilecek belirli kategoriler için etiketli Tweet veri setlerinin bulunmadığı görülmüştür. Bu çalışmada önerilen yaklaşımın uygulanacağı deneylerde kullanılmak üzere böyle bir veri setinin oluşturulması ihtiyacı doğmuştur. Öncelikle Twitter'dan tweet toplanıp daha sonra bu tweetler insanlar aracılığıyla manuel olarak etiketlenmiştir. Bigailab-5tweet-35K veri setinin oluşturulmasıyla ilgili bilgiler Bölüm 4.3.1'de verilmiştir.

Veri setleri arasında bir zaman korelasyonu gözetilmemiştir. Zaman, yazar (haberler), haber kaynağı, kullanıcı (Twitter) gibi faktörler rastgele seçilerek olabildiğince genel bir veri kümesinin oluşturulması amaçlanmıştır.

4.3.1. Twitter Verilerinin Toplanması

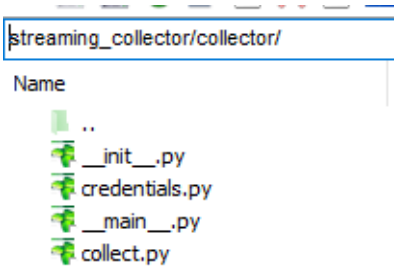
Sosyal medya analizinde, büyük miktarlarda farklı konu, kişi, tarih, popülerlik gibi çeşitliliği olan verilere ihtiyaç duyulmaktadır. Veri toplama işlemi için gerçekleştirilecek yazılımsal çözümün bu ihtiyaçlara cevap verebilecek şekilde olması gerekir. Tez kapsamındaki veri ihtiyaçları göz önünde bulundurularak Twitter'dan otomatik olarak nasıl veri toplanabileceği araştırılmıştır. Veri toplama seçenekleri detaylı bir şekilde incelenip, bu verileri otomatik olarak toplayan programlar kodlanmıştır.

Twitter'dan veri toplayabilmek için ilk koşul Twitter'a kayıt olarak bir kullanıcı hesabı oluşturmaktır. Veri toplama ve Twitter ile programsal olarak etkileşim kurmayı sağlayan diğer birçok hizmet (direkt mesaj, reklam, etkileşim, hesap aktivitesi API'leri vb.) geliştirici hizmetleri (https://developer.twitter.com) üzerinden sağlanmaktadır. Twitter verileri 2 farklı API kanalı yoluyla sunulmaktadır. Bunlar Rest API ve Streaming API'dir. Rest API ile önceki bir tarihe ait tweetler toplanabilmektedir. Sorgulara çoğunlukla son 7 gün içerisinde paylaşılmış tweetlerle cevap verilir. Streaming Api ile ise tweetler paylaşıldığı anda toplanmak üzere sunulmaktadır.

Twitter apileri ile veri toplamak için api hizmetlerini yazılım kütüphanesi olarak kullanıma açan açık kaynaklı kütüphaneler araştırılmıştır. Twitter4J (Java), Python-twitter (Python) ve Tweepy (Python) kütüphaneleri ile veri toplama işlemleri yapılarak detaylı olarak incelenmiştir. Bu çalışmadaki veri toplama işlemleri için gerekli programların geliştirilmesinde Tweepy kütüphanesinden yararlanılmasına karar verilmiştir.

Tweeter verilerinin JSON formatındaki yapıları irdelenmiştir. Bazı önemli veri anahtarları: *retweet_count, source, favorite_count, geo, text, place, in_reply_to_status_id, timestamp_ms, retweeted* şeklindedir. Metin sınıflandırma ve duygu analizi gibi problemlerde verinin sadece "text" değeri önemli iken eğilim analizi gibi problemlerde "retweet_count", "favorite", "geo" gibi bilgiler de önemli olabilmektedir. Verilerin herhangi bir şekilde küçültülmeden, dönen sonuçların olduğu gibi kaydedilmesi sosyal medya analizi sistemlerinde tercih edilmektedir. Bu nedenle verilerin saklanmasında doküman tabanlı, açık kaynaklı bir NoSQL veri tabanı sistemi olan MongoDB tercih edilmiştir.

Stream API ve Rest API üzerinden veri toplama işlemlerinin otomatikleştirilmesi için kabuk programı (bash) ve Python kodları bir arada kullanılarak basit birer yazılım geliştirilmiştir. Stream API için program dışardan verilen anahtarlara (kelime, hashtag veya kullanıcı adı) ve tarih, yer gibi diğer kriterlere uyan tweetleri toplamak üzere Twitter Stream Api ile bir bağlantı kurar. İstenilen süre boyunca veriler dinlenir ve kriterlere uyan veri oluştukça Twitter tarafından otomatik olarak gönderilir. İstemci-sunucu mekanizmasıyla çalışan bu programa ait temel kod dosyaları aşağıdaki Şekil 4.1'de verilmiştir.



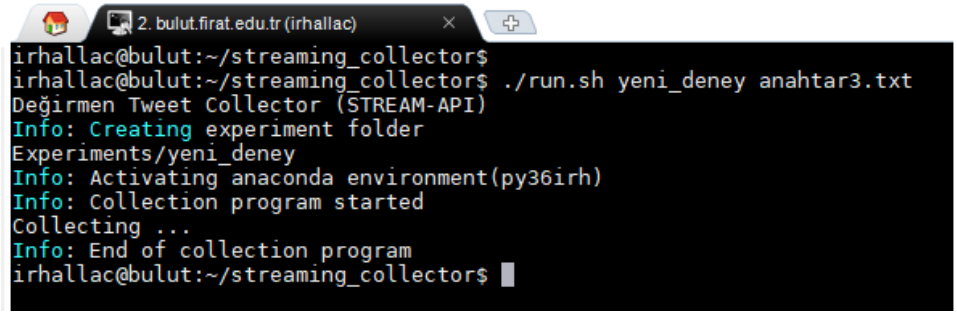
Şekil 4.1. Geliştirilen Twitter veri toplama aracının temel bileşenleri

Kabuk programı ile (run.sh) programın yürütülmesi yönetilmektedir. Tablo 4.1'de program özellikleri özetlenmiştir. Programın yürütülme adımlarına ait bir ekran görüntüsü Şekil 4.2'de gösterilmiştir. Program en fazla 10 saatlik periyotlar halinde test edilmiştir. 10 saat içerisinde 2 milyon adete kadar tweet toplandığı gözlenmiştir. Twitter'den gelen veri sayısının 200'e ulaştığı her periyotta sonuçlar yeni bir dosyaya kaydedilmektedir. Veri toplama deneylerinden sonuç dizinine ait örnek bir ekran görüntüsü Şekil 4.3'te verilmiştir. JSON

formatında dosyalara kaydedilen veriler daha sonra MongoDB ortamına aktarılmıştır. Rest API ile veri toplama programı da aynı yapıda tasarlanmıştır.

Tablo 4.1. Stream API ile veri toplama programı kullanımı

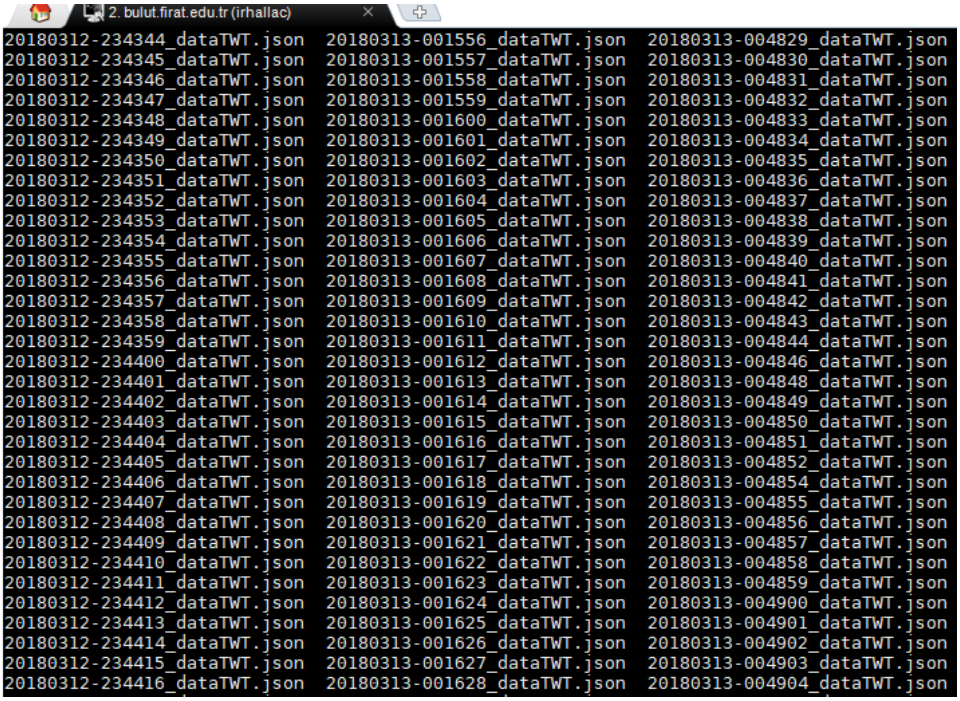
Program parçası	Görevi
run.sh	Tweetlerin nereye kayıt edileceği, arama kriterleri, arama süresi ve log dosyası parametrelerini kullanıcıdan alır. Gerekli kontrolleri yapar. Bunlar; kayıt dizini doğrulaması, arama kriterleri dosyasının doğrulanması, bu program için gerekli python Environment'ın aktif edilmesidir. Ayrıca sonuçların kaydedileceği dizinde arama dosyalarını yedekler.
credentials.py	TWITTER_APP_KEY, TWITTER_APP_SECRET, TWITTER_KEY ve TWITTER_SECRET bilgilerini kullanarak Twitter ile istemci-sunucu bağlantısını gerçekleştirir.
collect.py	Veri toplama işlemini yapan asıl programdır. Tweepy kütüphanesinin "StreamListener" sınıfını kullanır (overwrite).
__main__.py	Gerekli python kütüphanelerini yükler, Twitter bağlantısını test eder ve ana collect.py programını çağırır.



Şekil 4.2. Veri toplama programı ekranı

Tweetlerin rastgele toplanması, veri etiketleme aşamasındaki süreci karmaşıklaştıracağı için öncelikle istenilen kategorilere ait verilerin bulunabileceği bir veri arama stratejisi belirlenmiştir. Bu stratejide ekonomi, kültür, teknoloji, politika ve spor alanlarından tweet paylaşma ihtimali yüksek olan popüler Twitter kullanıcı hesapları belirlenmiştir. Örneğin ekonomi kategorisi için ekonomistlerden ve farklı yayın organlarının Twitter hesaplarından oluşan bir liste oluşturulmuştur. Benzer şekilde politika kategorisi için farklı siyasi partilerin liderleri ve milletvekillerinin hesaplarından bir liste oluşturulmuştur.

Belirlenen profillerin sadece kendi oluşturduğu tweetler kaydedilmiştir. Başka bir tweetin tekrar paylaşıldığı (RT işlemi) ve cevap olarak paylaşılan tweetler elenmiştir. Daha sonra, basit bir arayüz ile veriler etiketlemeye sunulmuştur. Mobil ve web tabanlı çalışan etiketleme uygulamasında tweet metnine uygun kategorinin kullanıcı tarafından seçilmesi sağlanmıştır. Belirlenen kategorilerden herhangi birine uygun olmayan metinler ise veri tabanından silinmiştir. Etiketleme sürecinde öğrenci ve akademisyenlerden oluşan yaklaşık 10 kişi görev almıştır. Her kategoriden 7000 adet veri etiketlenerek toplam 35 bin tweetten oluşan Bigailab-5tweet-35K veri seti oluşturulmuştur. Veri setine ait örnek tweet metinleri Tablo 4.2'de verilmiştir.



Şekil 4.3. Veri toplama çıktı ekranı

Tablo 4.2. Bigailab-5tweet-35K örnek tweet metinleri

Kategori	**Örnek Tweet**
Ekonomi	Çin ekonomisinin 6 aylık büyüme oranı açıklandı
Siyaset	Başbakanımız Sn. Binali Yıldırım, Ankara Etimesgut'ta vatandaşlara hitap ediyor.
Kültür	Kitapla oyun olur mu hiç? Okuma Atölyelerimiz oyunla kitabı bir araya getirmeye devam ediyor...
Spor	Fatih Terim: Bugün beni üzen Galatasaray ruhunun sahada olmayışıdır.
Teknoloji	Yapay zekâ fotoğraftan yemek tarifi çıkarıyor

4.4. Model

Yöntem olarak farklı derin öğrenme modelleri kullanılmıştır. Bu modeller MLP, CNN ve hibrit mimarili BiLSTM-CNN'dir. Modellerle ilgili detaylı literatür bilgisi Bölüm 3'te verildiği için bu bölümde ayrıca anlatılmamıştır. Farklı derin öğrenme model ve mimarilerinin metin sınıflandırma probleminde öğrenme aktarımı (transfer-learning) ve ince ayar (fine-tuning) teknikleriyle gösterdiği başarı incelenmiştir.

Bütün modellerde ortak olarak bir kelime gömme katmanı (embedding layer) kullanılmıştır. Bu katmanda haber veri seti üzerinden oluşturulan 10.000 boyutlu sözlük için her kelimeye karşılık bir kelime gömme değeri bulunmaktadır. Kelime gömmesi için başlangıç değerleri önceden eğitilmiş bir kelime gömme modeli olan FastText'ten [110] alınmıştır. Sözlükte yer almayan diğer bütün kelimeleri temsil etmek üzere "UNK" kelimesi (token) kullanılmıştır ve başlangıç değeri olarak bütün boyutlarına sıfır değeri atanmıştır. Gömme modeline ait önemli bazı özellikler Tablo 4.3'te verilmiştir.

Tablo 4.3. FastText gömme modeli özellikleri

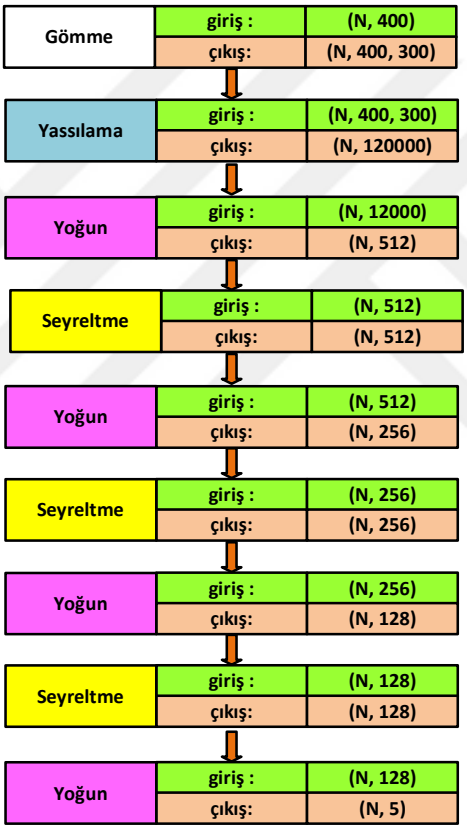
Özellik	Açıklama
Eğitimde kullanılan derlem ve dil	Türkçe Wikipedia dokümanları
Vektör boyutu	300
Yöntem	N-gram karakter dizilerine uygulanan Skip-gram tabanlı bir algoritma

Haber veri setinin en yüksek frekansa sahip 10.000 kelimesinin 9046'sı hazır kullanılan kelime gömme modelinde de bulunmaktadır. Geri kalan 954 kelimenin başlangıç değeri olarak yine sıfır vektörleri kullanılmıştır. Kelime gömme katmanı, bütün modellerde ağırlıkları eğitim sırasında güncellenecek (trainable) şekilde ayarlanmıştır. Bu şekilde bütün kelimelerin 300 boyutlu vektör ağırlıkları eğitim süresince iyileştirilmiştir.

Bu çalışmada yer alan derin öğrenme modellerin implementasyonunda Keras kullanılmıştır. MLP modelinde nöron sayıları sırasıyla 512, 256 ve 128 olan üç adet yoğun katman (dense-layer) kullanılmıştır. Aşırı öğrenmenin (over fitting) önüne geçmek için bu katmanlar arasına seyreltme (drop-out) katmanları yerleştirilmiştir. Son olarak çıkış katmanında sınıf sayısı (5) kadar nörondan oluşan bir yoğun katman kullanılmıştır. MLP modelinin mimarisi detaylı olarak Şekil 4.4'te gösterilmiştir. Uygulanan iyileştirme adımlarının farklı derin öğrenme modelleri üzerinde gösterdiği performansın incelenmesinde kolaylık sağlaması amacıyla MLP modeli özellikle basit tutulmuştur. Bu şekilde model karmaşıklığının yüksek olmaması sayesinde

yüksek başarı elde edilmesi hedeflenen hibrit modele paralel olarak başarının artıp artmadığı gözlemlenerek bir çeşit sağlama adımı olarak kullanılmıştır.

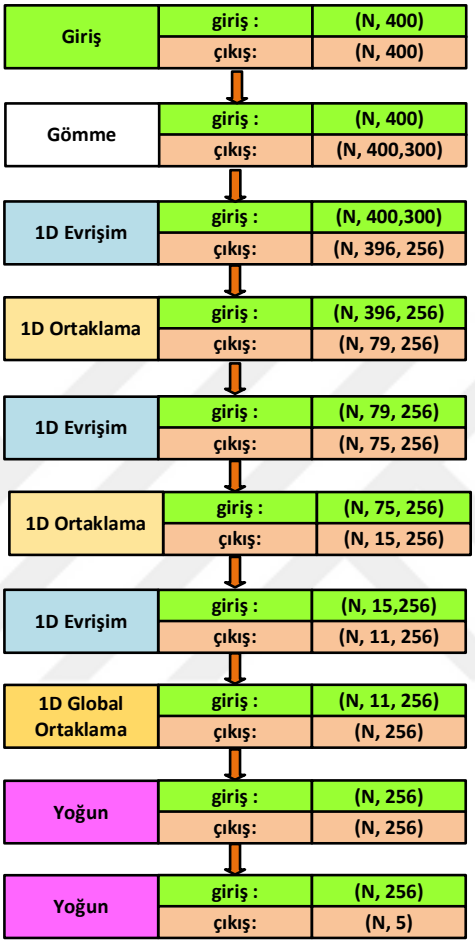
Kullandığımız CNN modelinin bu uygulamasında genellikle standart olarak uygulanan bir dizi evrişim ve ortaklama katmanları kullanılmıştır. Evrişim adımlarının tamamında filtre sayısı 256, kernel boyutu 5X5 ve aktivasyon fonksiyonu ReLu olarak seçilmiştir. Bu adımlar sonunda 256 nörondan oluşan bir yoğun katman kullanılarak CNN adımlarında öğrenilen özelliklerin sınıflandırma öncesinde birdenbire kaybedilmemesi hedeflenmiştir. Modelin çıkışında, diğer tüm modellerde olduğu gibi 5 nörondan oluşan bir yoğun katman kullanılmıştır. Bu modelin detayları Şekil 4.5'te gösterilmiştir.



Şekil 4.4. Tweet sınıflandırma için MLP modeli

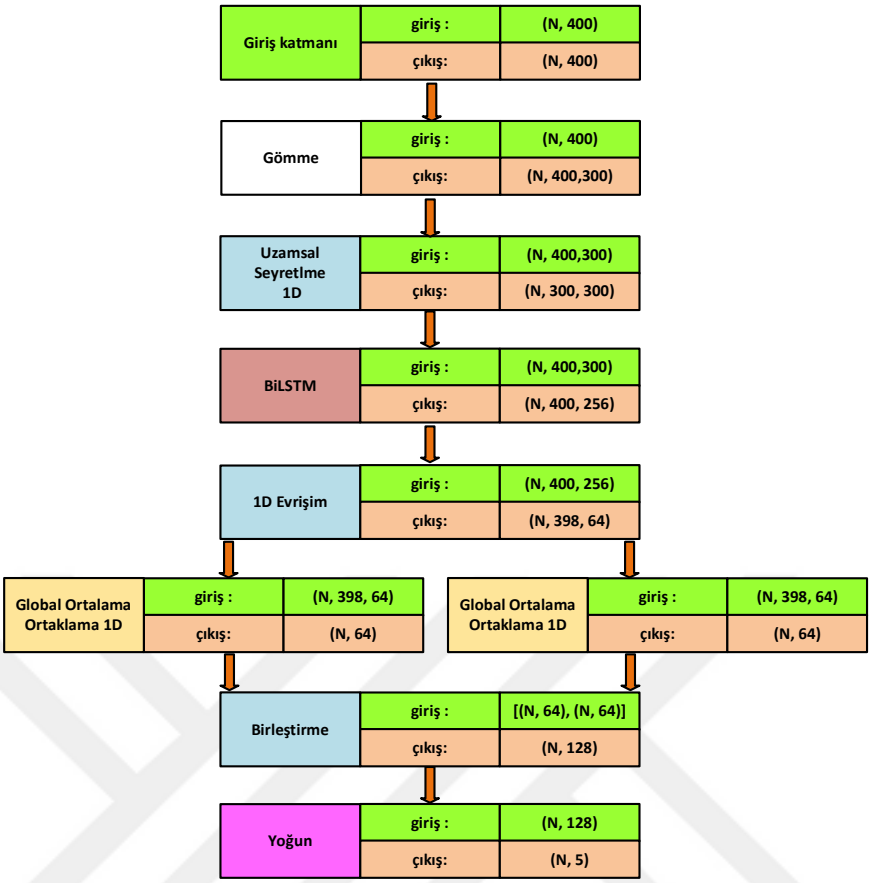
Bu problemde birçok farklı derin öğrenme mimarisinin denenmesi yerine temel mimarilerden iki tanesiyle birlikte üçüncü model olarak tez kapsamında gerçekleştirilen duygu sınıflandırma çalışmasında ve birçok doğal dil işleme probleminde yüksek başarı gösteren Bi-LSTM ve CNN modellerinin arka arkaya kullanıldığı hibrit bir derin öğrenme modeli kullanılmıştır [111,112]. Bu modelin mimarisinde, BiLSTM yerine farklı RNN varyantlarının kullanıldığı metin sınıflandırma çalışmaları da mevcuttur [113,114]. Bu modellerde ayrıca karakter veya kelime tabanlı farklı gömme katmanlarının kullanıldığı görülmektedir. Kısaca

BiLSTM-CNN olarak adlandırdığımız bu modelde metnin sağdan sola ve soldan sağa temsillerinin elde edildiği iki yönlü LSTM katmanlarının sonrasında elde edilen bu özelliklerin CNN katmanına aktarılması ve sonrasında sınıflandırma işleminin softmax fonksiyonu ile uygulanması amaçlanmıştır.



Şekil 4.5. Tweet sınıflandırma için CNN modeli

Kelime gömme katmanı ile BiLSTM katmanlarına geçiş öncesinde 0.35 oranlı seyreltme (dropout) katmanı kullanılmıştır. BiLSTM'nin gizli katmanındaki düğüm sayısı 128 ve seyreltme oranı 0.15'tir. BiLSTM sonrasında uygulanan evrişim katmanında filtre sayısı 64, kernel boyutu 3X3 olarak belirlenmiştir. Evrişim katmanı çıkışında ise ortalama alan (Global Average Pooling) ve en büyüğünü alan (Global Max Pooling) 2 ayrı ortaklama (pooling) katmanı uygulanmıştır ve bu katmanlar birleştirilerek 128 nöronlu yoğun katmana bağlanmıştır. Hata fonksiyonu olarak ise bütün modellerde ortak olarak kategorik çapraz entropi (categorical-cross-entropy) kullanılmıştır. Optimizasyon algoritması olarak Adam [115] yöntemi tercih edilmiştir. Model mimarisinin ayrıntıları Şekil 4.6'da gösterilmiştir.



Şekil 4.6. Tweet sınıflandırma için hibrit BiLSTM-CNN modeli

4.5. Öğrenme Aktarımı Yaklaşımı

Bu bölümde, derin öğrenme modellerinin metin sınıflandırma probleminde gösterdiği başarının incelenmesi ve alan adaptasyonu amacıyla gerçekleştirilen ince ayar yöntemi anlatılmıştır.

Bölüm 4.4'te açıklanan derin öğrenme yöntemlerinin haber veri seti ile eğitildikten sonra aynı kategorilere göre gruplanmış tweet metinleri üzerinde gösterdiği sınıflandırma başarıları değerlendirilmiştir. Modellerin performansları Tablo 4.4'te verilmiştir.

Tablo 4.4. İnce ayar yöntemi öncesi model performansları

Model	Başarı oranı (%)	
	Haber metinleri üzerinde	Tweet metinleri üzerinde
MLP	95.62	52.14
CNN	95.85	**56.97**
BiLSTM-CNN	**96.54**	53.68

Haber modellerinin eğitimi için kullanılan Bigailab-5news-500K veri seti; %80 eğitim ve %20 doğrulama şeklinde 2 parçaya ayrılmıştır. Haber modellerinin, doğrulama verisi üzerinde en iyi performans gösteren versiyonlarına ait ağırlık değerleri kaydedilmiştir. Tablo 4.4'te verilen "Haber metinleri üzerinde" başarı değeri; her kategoriden eşit sayıda haberin yer aldığı 100.000 adet haber doğrulama verisi için elde edilen başarıyı göstermektedir. Ayrıca bir haber test veri seti kullanılmamıştır. Bigailab-5tweet-35K veri seti ise modellerin ince ayar adımlarında kullanılmak üzere rasgele seçim yapılarak eğitim, doğrulama ve test olarak 3 parçaya ayrılmıştır. Verilerin rasgele seçim işleminde sınıflar arasında eşit sayıda veri bulunması gözetilmiştir. Bu şekilde hem eğitim hem de test aşamalarında dengeli veri setleri kullanılmıştır. Tablo 4.5'te de görüleceği üzere manuel etiketlenmiş tweet veri setinin %70'inden fazlası modellerin testinde kullanılmıştır. Veri setlerinin kullanıldığı aşamalara ait dağılımlar Tablo 4.5'te verilmiştir.

Tablo 4.5. Haber ve tweet veri setlerinin kullanım dağılımı

İşlem	Sayı
Haber eğitim	400.000
Haber doğrulama	100.000
Tweet eğitim	8.000
Tweet doğrulama	2.000
Tweet test	25.000

Tablo 4.6. İnce ayar (fine-tuning) adımları

Adım-1	**1-** Haber dokümanlarıyla eğitilen modeli en iyi ağırlıklarıyla yükle.
	2- Son katman (dense layer) haricindeki bütün katmanları dondur.
	3- Modeli tweet verileriyle eğit, en iyi ağırlıkları kaydet.
Adım-2	**1-** Önceki adımda eğitilen modeli yükle.
	2- Daha fazla katmanı eğitime aç.
	3- Modeli yeniden tweet verileriyle eğit, en iyi ağırlıkları kaydet.
Adım-3 (son adım)	**1-** Önceki adımda eğitilen modeli yükle.
	2- Bütün katmanları eğitime aç.
	3- Modeli tweet verileriyle eğit, en iyi ağırlıkları kaydet.
Adım-4 (test)	**1-** Haber dokümanlarıyla eğitilen modeli en iyi ağırlıklarıyla yükle.
	2- Bütün katmanları eğitime aç.
	3- Modeli tweet verileriyle eğit, Adım-3'teki model ile karşılaştır.

Bir derin öğrenme modelinin büyük miktarda haber verisiyle eğitildiği ilk adımdan sonra uygulanan ince ayar adımlarının detayları Tablo 4.6'da verilmiştir. Bir katmanın dondurulması, o katmanda bulunan nöronlara ait ağırlık değerlerinin eğitim sırasında değiştirilmesine izin verilmemesi demektir. Tersi durumda ise ağırlıklar yeni gelen verilere göre güncellenecektir.

4.6. Sonuçlar

Tablo 4.4'te öğrenme aktarımı yaklaşımının uygulanması öncesinde MLP, CNN ve BiLSTM-CNN modellerinin haber ve tweet metinlerini sınıflandırmada elde ettiği doğruluk değerleri gösterilmiştir. BiLSTM-CNN'nin haber sınıflandırmada en iyi başarıyı gösterdiği görülmüştür. Ancak hiçbir tweet verisiyle karşılaşmayan bu 3 model arasında en iyi tweet sınıflandırma performansını CNN göstermiştir.

Modellere Tablo 4.6'da verilen Adım-1, Adım-2 ve Adım-3 öğrenme aktarımı adımları uygulandığında elde edilen tweet sınıflandırma başarıları Tablo 4.7'de verilmiştir.

Tablo 4.7. İnce ayar yöntemi sonrası model performansları

Model	**Başarı oranı (%)**			
	Adım-1	**Adım-2**	**Adım-3**	**Adım-4**
MLP	68.70	80.46	85.11	84.43
CNN	71.04	81.26	83.92	83.72
BiLSTM-CNN	86.04	87.24	**87.97**	87.20

Uygulanan iyileştirme adımlarının tweet sınıflandırma başarısını arttırdığı Tablo 4.7'de her üç model için de net bir şekilde görülmektedir. Her bir model için elde edilen toplam doğruluk artışı Tablo 4.8'de verilmiştir. En yüksek %63.87 değeriyle en büyük artışı BiLSTM-CNN modeli göstermiştir.

Tablo 4.8. İnce ayar yöntemi sonrası görülen doğruluk artış miktarları

Model	**Başarı artışı (%)**
MLP	63.23
CNN	47.30
BiLSTM-CNN	**63.87**

Doğrudan Adım-4'ün uygulanmasıyla elde edilen sonuçların önerilen iyileştirme yaklaşımından daha iyi olmadığı ıspatlanmıştır. Bununla birlikte, Adım-4'ün yine de haber modellerinin tweet sınıflandırma başarısını büyük oranda arttırdığı görülmüştür.

4.7. Değerlendirme

Bu çalışmada, az miktarda etiketli tweet olduğu fakat farklı bir kaynaktan nispeten büyük miktarlarda etiketli veri olduğu şartlarda tweetlerin yüksek doğruluklarda sınıflandırılmasını gerçekleştiren, derin öğrenme modelleri üzerinde uygulanabilecek bir alan adaptasyonu yaklaşımı önerilmiştir.

BiLSTM-CNN öğrenme mimarisinin bir yandan dile ait semantik özellikleri öğrenirken, bir yandan da amaç fonksiyonun yönlendirdiği ilgili verinin ait olduğu kategoriye ait özellikleri öğrenmesinin mümkün olduğu görülmüştür.

Derin öğrenme modellerinin iyileştirilmesinde, yeni verilerle doğrudan bir eğitim gerçekleştirmek yerine sınıflandırıcı katmanından geriye doğru katmanların aşama aşama öğrenmeye açılmasının daha yüksek başarı sağladığı gösterilmiştir.

Önerilen bu yaklaşım sonunda elde edilen en iyi sonucu veren model kullanılarak genel bir tweet sınıflandırıcısı ortaya konulmuştur. 5 kategoriden 35 bin adet tweet verisi etiketlenmiştir. Manuel olarak açıklamalı bir tweet veri seti ve bir haber veri seti koleksiyonunun Türkçe için yayınlanması bu alandaki gelecek çalışmalara katkı sağlayacaktır.

5. DAĞITIK DOKÜMAN VEKTÖRLERİYLE SOSYAL MEDYA KULLANICI TEMSİLİ

Son yıllarda sosyal medya verilerinden yararlı bilgilerin çıkarılması ve analiz edilmesi konularında yapılan çalışmaların araştırmacılar arasında giderek artan bir ilgiyle devam ettiği görülmektedir. Sosyal medya analizi konusundaki bazı önemli problemler ve çalışma alanları şu şekildedir:

- Duygu analizi (sentiment analysis)
- Tavsiye sistemleri (recommendation systems)
- Özetleme (summarization)
- Konu tespiti (topic detection)
- Anahtar kelime çıkartımı (keyword detection)
- Etki analizi (influence analysis)
- Kümeleme (clustering)
- Sınıflandırma (classification)
- Benzerlik tespiti (similarity detection)
- Topluluk tespiti (community detection)
- Memnuniyet analizi (satisfaction analysis)
- Varlık tespiti (entity identification)
- Okunabilirlik analizi (readability analysis)
- Örüntü tespiti (pattern detection)
- Eğilim tespiti (trend identification)
- Kullanıcı temsili, ürün temsili vb.

Veri madenciliği ve makine öğrenmesi yöntemleri dışında basit istatistik yöntemler kullanılarak gerçekleştirilen çalışmalar da bulunmaktadır. Bunlara örnek olarak aşağıdaki analizler verilebilir:

- Gündem analizi
- Popüler hesapların tespiti ve analizi
- Zamana göre takipçi sayısının değişimi
- Aktivite analizi
- Bir paylaşımın zamana göre yayılımı
- Hashtag veya anahtar kelimeye göre paylaşım trafiklerinin karşılaştırılması
- Paylaşımların sanal topluluklara göre dağılımları
- Konum tabanlı dağlımlar
- Cinsiyet, yaş, dil vb. dağılımlar

- Paylaşımların yapıldığı cihazların özelliklerine göre dağılımlar: Telefon / PC / Linux / Windows / iOS / Android vb.
- Ayrıca, "X kişisi hakkında en çok kimler konuşuyor?", "X konusu hakkında en çok kimler konuşuyor?" gibi sayısal verilerden yararlanılabilir.

Tezin önceki bölümlerinde (Bölüm-3, Bölüm-4) de gösterildiği üzere kullanıcı tarafından oluşturulan içeriklerin analiz edilmesi bakımından farklı çalışma alanları ve yöntemler mevcuttur. İçeriklerin ana üretim kaynağı olan sosyal medya kullanıcılarının vektörel temsillerinin oluşturması, bu zengin veri deryasından istifade etmeye yönelik geliştirilen birçok uygulamanın kilit unsurlarından birisidir. Bu bölümde, bir sosyal ağın kullanıcıları hakkında otomatik olarak anlamlı gözlemler oluşturmada kullanılabilecek, kullanıcılar için gerçek değerli vektörler üretebilen kullanıcı temsil modellerinin geliştirilmesi konusunda gerçekleştirilen çalışmalar anlatılmıştır.

Bu bölümde gerçekleştirilen çalışmalar "user2Vec: Social Media User Representation Based on Distributed Document Embeddings" isimli bildiri ile sunulmuştur [116].

Bu tez çalışması kapsamında Twitter örneği üzerinden sosyal medya kullanıcılarının vektör uzayındaki temsillerine karşılık kullanılan; kullanıcı temsili (user representation), kullanıcı gömmesi (user embedding) ve kullanıcı vektörü (user vector) terimleri aynı anlamı taşımaktadır ve yer yer birbirlerinin yerine kullanılmıştır.

5.1. Problemin Tanımı

Sosyal medya kullanıcıları, çevrimiçi bir platform üzerinde oldukça çeşitli ilgi alanlarıyla ilgili içerikler paylaşmaktadır. Örneğin, bir kullanıcı Twitter üzerinde bir hesap oluşturduğunda kendisinden bazı temel ilgi alanları konusunda bilgi girişi yapması istenmektedir. Kullanıcının politika, teknoloji, spor gibi seçeneklerden bazılarını tercih etmesi sonrasında özellikle o alanlarla ilgili konularda ilgisini çekebilecek öne çıkan kişileri takip etme önerileri sağlanmaktadır.

Bir siyasi parti liderini siyaset konusuyla ilişkilendirmek veya spor gazetesinde köşe yazıları yazan bir kişiyi spor konusuyla ilişkilendirmek oldukça anlaşılır eşleştirmelerdir. Fakat benzer ilişkilendirmelerin genel olarak bütün Twitter kullanıcıları için gerçekleştirilmek istenmesi durumunda bu problem oldukça karmaşık bir hal alacaktır. Çünkü, sosyal medya kullanıcıları genellikle oldukça geniş bir yelpazedeki içeriklerle etkileşim kurmaktadır ve spesifik konularla ilişkilendirilmeleri zordur. Fakat kullanıcılar için vektörel uzayda temsiller (gömme) elde edilirken bu geniş yelpazeden ilişkilerin olabildiğince gömülü olması gereklidir.

Bu çalışmada önerilen kullanıcı gömme modeli ile bir kullanıcının farklı konularla olan ilişkileri tek bir vektör içerisinde temsil edilmektedir. Kullanıcıların önceden tanımlı gruplarda etiketlenmesine veya belirli sayıda kümede yer almasına ihtiyaç duymadan, yoğun vektörlere

dönüştürülerek birçok farklı uygulamada yararlanabilecek bir temsil elde edilmektedir. Bu sayede kullanıcıların hem birbirine benzer hem de birbirinden farklı olan karakteristikleri sayısal dünyada temsil edilerek kullanıcılar hakkında sayısız özelliklerin keşfedilmesine olanak sağlanmaktadır.

Kelime temsil (word-embeddings) yöntemlerindeki son gelişmeler çok sayıda metin tabanlı bilgi getirimi (information-retrival) uygulamasının gelişimine katkı sağlamıştır. CBOW ve Skip-Gram algoritmaları, düşük boyutlu vektör uzaylarındaki kelimelerin birçok semantik karakteristiğini başarıyla temsil eden metinsel temsil yöntemleri konusunda yeni teknikler ortaya çıkarmıştır. Bu çalışmada, sosyal medya kullanıcılarını temsil etmek için doküman temsil modeli (doc2vec) kullanarak yapay sinir ağı tabanlı bir kullanıcı temsili (user embedding) modeli ortaya konulmuştur.

Kullanıcı vektörlerinin kalitesini değerlendirmek için basit bir benzerlik testi yöntemi önerilmiştir. Elde edilen sonuçlar ayrıca ortalama kategori vektörleri şeklinde incelenerek kullanıcıların kümelenmesi ve topluluk keşfi problemlerine yönelik bir kullanım senaryosu gerçekleştirilmiştir. Çalışmanın sonucunda elde edilen *user2vec* modeliyle kullanıcıların semantik olarak anlamlı gösterimlerinin elde edildiği ve bu yöntemin çeşitli açılardan geliştirilmeye oldukça açık olduğu görülmüştür. Ayrıca ilerde önerilecek yeni yöntemlerin değerlendirilmesinde kullanılabilecek önemli bir veri seti literatüre sunulmuştur.

5.2. Literatür Özeti ve Önerilen Yaklaşım

Temsil öğrenme (representation learning), makine öğrenmesi yöntemlerinin uygulanmasındaki önemli aşamalardan biridir. Nesnelerin önemli niteliklerinin belirlenmesi ve temsillerinin elde edilmesinde kullanılan yöntemlerin birbirine göre farklı avantaj ve dezavantajları bulunmaktadır. Örneğin; yerel temsil (local representation) yöntemleri basit ve kolay uygulanabilir olmaları yönünden avantajlıdır [117]. İnsan beyninin hafıza mekanizmasından esinlenen dağıtık temsiller ise uygulamada bir takım zorlukları beraberinde getirir, ancak nesnelerin (öğelerin) mikro özelliklerini kodlayabilmesi yönünden yerel temsillerden daha iyidir.

DDİ uygulamalarında kelime gömmeleri için Mikolov ve arkadaşları tarafından önerilmiş iki temel yöntem bulunmaktadır [63]. Bunlar; CBOW ve Skip-Gram'dır. Bu yöntemler sayesinde doğal dile ait çeşitli özelliklerin ve örüntülerin büyük miktarda metin verisi üzerinden öğrenilmesi mümkündür. Makine çevirisi, soru cevaplama, metin sınıflandırma gibi birçok DDİ probleminin çözümünde bu yöntemlerden yararlanılmaktadır. Kelimelerin oluşturduğu cümle, paragraf veya doküman olarak nitelendirilen metinlerin gömmelerinin elde edilmesi için kelime gömme yöntemlerinin değiştirilmiş versiyonları önerilmiştir [118]. Genellikle paragraf veya doküman vektör modeli olarak adlandırılan bu nöral gömme modelleri ile farklı uzunluklardaki metinlerin

sabit-uzunluklu vektör değerleri elde edilir. Bu tekniklerin genel yaklaşımı; Skip-gram veya CBOW gibi bir kelime gömme yönteminin bir derleme uygulanması sırasında cümle, paragraf gibi daha farklı uzunluktaki birimlerin de kelime gibi ele alınarak temsillerinin elde edilmesi şeklindedir. Bu yöntemlerle ilgili gerekli ön bilgiler Bölüm 2'de verilmiştir.

Sosyal medya kullanıcıları üzerinde, çeşitli bilgi çıkartımı ihtiyaçlarının çözümüne yönelik gömme modellerinin oluşturulması son zamanlarda giderek büyüyen bir araştırma alanıdır. Kullanıcı temsilleri, cinsiyet tahmini [119] ve kişilik özelliklerini belirleme [120,121] gibi çeşitli uygulamalarda kullanılabilir. Bu çalışmaların önemli bir çoğunluğunda güncel kelime ve cümle temsil modellerinden esinlenildiği görülmektedir. Aynı şekilde kelime ve doküman temsil tekniklerinin metinsel olmayan verilerin kullanıldığı problem alanlarında da kullanılması yönünden artan bir eğilim söz konusudur. Örneğin; görüntü [122], biyolojik veriler [123,124], kullanıcı aktivite verisi [125], ürün ve kullanıcı gibi birçok farklı alandan verilerin özellik çıkarımında ve temsilinde yararlanılmaktadır [126].

5.2.1. İlgili Çalışmalar

Mishra ve arkadaşları, istismarcı (abusive) kullanıcıları tahmin etmek için kullanıcıların topluluk temelli bilgilerini kodlayan bir gömme metodolojisi önermiştir [127].

Yu ve arkadaşları, Paragraf Vektörü modeli yönteminden yararlanarak makine öğrenmesi konusunda çalışan araştırmacılara yönelik bir mikro blog sayfasının kullanıcıları için kişiye özel bir tavsiye öneri sistemi önermiştir [128]. İngilizce ve Çince içeriklerden oluşan verilerden yola çıkarak araştırmacıları temsil etmek üzere 50 ila 300 arasında değişen boyutlar için yoğun vektörler elde etmiştir.

Amir ver arkadaşları, Twitter verilerini kullanarak kullanıcıların ruh sağlığını tahmin etmek için önceden eğitilmiş bir kelime gömme katmanı ve gizli bir katman içeren ileri beslemeli bir sinir ağı önermiştir [129]. Jaech ve arkadaşları, topluluk üyesi tespiti (community member retrieval) görevi için benzer bir kelime torbası nöral modeli önermiştir [130].

Benton ve arkadaşları, Twitter kullanıcılarının belirli bir konudaki duruşlarının tespit edilmesi üzerine gerçekleştirdikleri görüş sınıflandırma (stance classification) çalışmasında yazarlar hakkında bağlamsal bilgileri kullanan GRU tabanlı bir model önermiştir [131].

Xing ve Paul, Twitter kullanıcılarının tweetlerinin bir araya getirilmesiyle elde edilen dokümanlara paragraph2vec algoritmasını uygulamıştır [132]. Veri setinin oluşturulması için Amerikan üniversitelerinin kullanıcı hesaplarını takip eden kullanıcılar arasından 5000 kişi rastgele seçilmiş ve bu kişilerin İngilizce dilinde yazılmış tweetleri toplanmıştır. Modelin başarısını arttırmak için kullanıcıların tweetleriyle; kullanıcı adı, konum ve takipçi bilgileri birleştirilmiştir. Çalışmada, konum bilgisi gibi normalde metinsel olmayan ayrık özelliklerin tweet metniyle birlikte düz bir şekilde birleştirildiği taktirde modelin başarısında bir artış

sağlanamadığı ve bu nedenle bu bilgilerin "dummy token" adı verilen kukla kelimelerle genişletilerek (augmented) kullanılması gerektiği belirtilmektedir. Yöntemin başarısının değerlendirilmesinde kosinüs benzerliği değerini metrik olarak kullanarak takipçilerinin tahmin edilmesi üzerinden bir doğrulama gerçekleştirilmiştir.

Amir ve arkadaşları, gizli alay (sarcasm) tespiti için CNN tabanlı bir yapay sinir ağının kullanıldığı yeni bir yaklaşım önermiştir [133]. Yapay sinir ağı modelinin eğitimi sırasında yazılan metinin ait olduğu kategori öğrenilirken bir yandan da o metnin yazarının kim olduğu bilgisinin öğrenilmesi amacıyla yazar gömmesi (author embedding) katmanı kullanılmıştır. Bu şekilde yazılan bir sosyal medya yorumunun gizli alay içerip içermediği tahmin edilirken yorumun kimin tarafından yazıldığı da dikkate alınarak sınıflandırmanın başarısında artış sağlanmıştır.

5.2.2. Metodoloji

Literatürdeki çalışmalar incelendiğinde, genel yaklaşımın; bir kullanıcının oluşturduğu bütün yazı içeriğinin (tweet, blog yazısı vb.) bir araya getirilip tek bir doküman olarak ele alınması ve bu dokümanın temsilinin kullanıcı temsili olarak kullanılması şeklinde olduğu görülmüştür. Bu modellerin gerçek hayattaki uygulamaları düşünüldüğünde verilerin bu şekilde toplanmış ve kullanıma hazır olduğu senaryoların gerçekleşme olasılığı zayıf olacaktır. Yani, kullanıcı verilerine model eğitimi aşamasında erişim konusunda çeşitli noksanlıklar ve imkansızlıklar olacaktır.

Sezgisel olarak, eğer kullanıcılar sosyal medyada yayınladıkları içerikler üzerinden temsil edilecekse öncelikle her bir paylaşımın (tweet, yorum, vb.) olabildiğince iyi temsil edilmesi gereklidir. Birbirleriyle alakalı olma gibi bir varsayımın yapılamayacağı binlerce tweetin bir araya getirilmesi ve bu dokümanlar üzerinden elde edilen vektörlerin kullanıcı temsili olarak kullanılması bu önseziye göre doğru bir yaklaşım olmayacaktır. Diğer kullanıcılara göre çok daha az sayıda tweet paylaşmış kullanıcıların da başarılı bir şekilde temsil edilmesi söz konusu olduğunda bu faktör başarıyı azaltacaktır. Bu nedenle, önerilen modelin öncelikle tweet temsilinde, daha sonra kullanıcı temsilinde iyi olması hedeflenmiştir. Bir doküman modeli eğitilerek yeni gelen her tweetin bu model üzerinden temsilinin elde edilmesi (inferring mechanism) mümkündür.

Literatürdeki kullanıcı gömme çalışmalarında öne çıkan problemlerden bir diğeri ise önerilen yöntemlerin başarılarının doğrulanma testleridir. Elde edilen vektörlerin kalitesinin test edilmesinde genellikle; X kullanıcısı Y kullanıcısına Z kullanıcısından daha yakın ise onu takip ediyor olmalıdır veya aralarında bir bağlantı olmalıdır gibi başarı ölçüt kriterlerinin uygulandığı görülmektedir. Bu yaklaşım, birbirini takip etmeyen, dağıtık bir kullanıcı kümesi üzerinde uygulandığında çeşitli problemler ortaya çıkacaktır.

Kelime vektörlerinin doğruluğu ölçülmek istendiğinde anlamca birbirine yakın kelimelerin semantik uzaydaki temsillerinin de bu benzerliğe paralel şekilde sayısal yakınlık gösterip göstermediğinin testi uygulanabilecek yaklaşımlardan birisidir. Benzer şekilde bu çalışmada da, önerilen modelin başarısını ölçmek üzere önceden belirlenen çeşitli kategorilere ait olacak şekilde Twitter kullanıcıları belirlenmiştir. Kullanıcı temsillerinin bu kategoriler bazında aynı gruplarda olacak şekilde vektör uzayında konumlanmalarına göre doğrulama testleri gerçekleştirilmiştir. 5 farklı semantik kategori belirlenerek her bir kategoriye uygun olarak 200'er tane Twitter kullanıcısı manuel olarak seçilmiştir. Belirlenen kullanıcılara ait tweetler toplandıktan sonra sadece İngilizce yazılmış olan tweetler kullanılarak bir derlem oluşturulmuştur. Bu derlem ile bir doküman temsil modeli eğitilmiştir. Doküman modeli kullanılarak her bir kullanıcı kendi yazmış olduğu tweetlerin temsil vektörlerinin ortalamasıyla temsil edilmiştir. Benzer şekilde belirlenen 5 kategori için de o gruba ait kullanıcıların temsillerinin ortalaması alınarak grup temsil vektörleri elde edilmiştir.

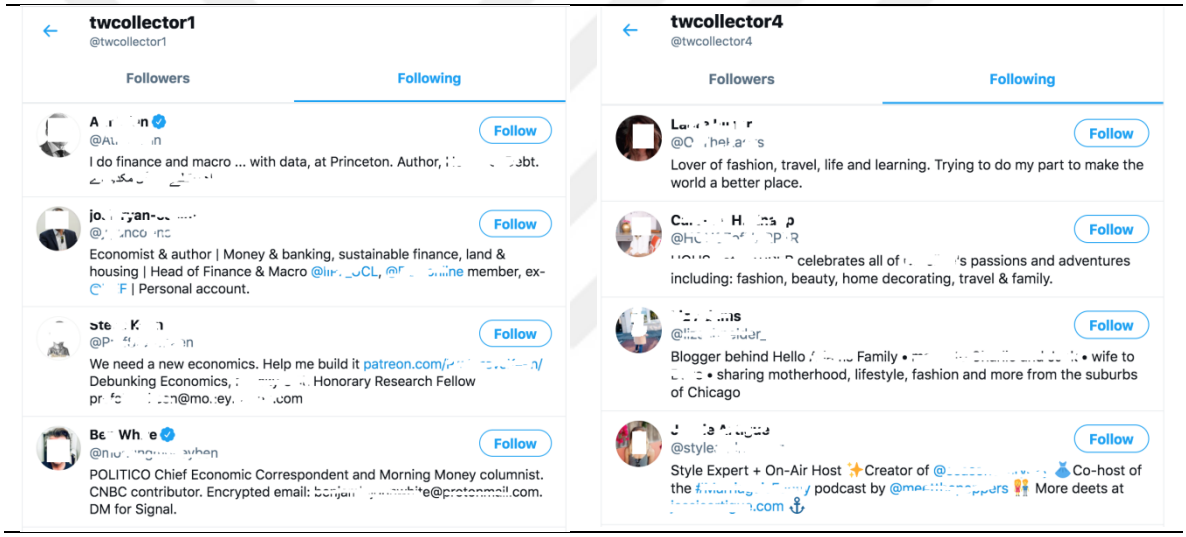
5.3. Veri Seti

Veri setinin oluşturulması bu çalışmanın önemli bir parçasıdır. Önerilen modelin test edilmesi amacıyla gerçekleştireceği tahminlerin manuel hazırlanmış kullanıcı gruplarıyla karşılaştırılması fikri üzerine altın standart bir kullanıcı veri seti hazırlanmıştır.

Bir kişi kısa bir süre için bazı sosyal medya profillerini incelediğinde o profillerin konuştuğu konuları, takip ettiği, beğendiği ve farklı şekillerde (beğeni, yorum, re-tweet) desteklediği kişileri göz önünde bulundurarak o sosyal ağda bulunan hangi profillerin birbirine daha benzer bir varlık gösterdiğine dair bir kanıya varabilir. Hangi kullanıcıların benzer olarak değerlendirilmesi gerektiğine karar vermek sübjektif bir görev olsa da, bu görüşler arasında genellikle bir uzlaşma olacaktır. Altın standart için ideal olan, manuel seçimleri yapan insanlar arasında iyi düzeyde bir anlaşmanın olmasıdır.

Verilerin elde edilmesinde öncelikle ekonomi, kripto-ekonomi, teknoloji, siyaset ve moda olmak üzere 5 konu başlığı seçilmiştir. Her bir kategoriye ait verilerin toplanması için ayrı bir Twitter kullanıcısı oluşturulmuştur. Şekil 5.1'de bu kullanıcılara ait profil sayfalarının ekran görüntüleri gösterilmiştir. Şekil 5.2'de ise örnek olarak takip edilen bazı kullanıcılar gösterilmiştir. Bu 5 kullanıcı üzerinden ilgili kategoride olması uygun olacak kullanıcıların seçimi manuel olarak gerçekleştirilmiştir. Çeşitli internet kaynaklarından "Ekonomi alanında takip edilmesi gereken en önemli 10 kişi …" gibi makaleler incelenip başlangıç olarak az sayıda kullanıcı seçildikten sonra Twitter'ın yaptığı otomatik "takip-et" önerileri de dikkate alınarak 5000'den fazla profil incelenmesi sonucunda 5 kategori için toplam 1000 profil seçilmiştir. Manuel olarak seçilen kullanıcılar yöntemin test edilmesinde altın standart (gold-standard-dataset) olarak kullanılmıştır.

Şekil 5.1. Kullanıcı listesi oluşturmak için açılan Twitter hesapları



Şekil 5.2. Örnek Twitter kullanıcı listeleri

	tweet_id	user_name	text	category
0	106531841 4128	Ba .Obi na	I am grateful for the next generation of leade...	twcollector5
1	10650565 19296	Ba kOb ma	Thanks to the Chicago @FoodDepository team for...	twcollector5
2	10643002323 408	Ba bOb na	When someone shares their story, we see the wo...	twcollector5
3	10 44762105857	Ba kOb ma	Our future depends on all our young people, in...	twcollector5
4	106 184938377216	Ba :Ob ma	Of course, @MichelleObama's my wife, so I'm a ...	twcollector5
...	...	...	...	...
995	87180613983 17	u a wi z	Pointing out now is not 1987 doesn't mean 'eve...	twcollector1
996	87179 374560	u a w z	1987 vs now $SPX https://t.co/9RjFFKVAvT	twcollector1
997	87 4672846065664	u w z	Kahneman and Tversky "availability heuristic":...	twcollector1
998	8714 0364975104	u ar z	While an interesting chart, don't lose sight o...	twcollector1
999	87138967 138368	u. w z	When this ratio is low (like now), $SPX has: -...	twcollector1

500000 rows × 4 columns

Şekil 5.3. Twitter veri setine genel bakış

Kullanıcı listeleri oluşturulduktan sonra Twitter REST API üzerinden otomatik olarak tweet toplanmıştır. Çalışmanın gerçekleştirildiği tarihlerde REST API bir kullanıcının en fazla son 3.200 tweetinin erişimine izin vermiştir. Veri toplama aşamasında bu üst limite bağlı kalınmıştır. Profillerdeki İngilizce olmayan metinler, resimler ve retweetler veri setine dâhil edilmemiştir. Bu nedenle, her bir kullanıcının ve kategorideki toplam tweetlerin sayısında bir dengesizlik ortaya çıkmıştır. Veri dengesizliği etkisinin önerilen yönteme yansımaması için her gruptan rastgele 100 kullanıcı seçilerek bu kişilerin en güncel olan 1000'er adet tweeti alınmıştır. Sonuç olarak 500 kullanıcıya ait toplam 500.000 tweetten oluşan bir veri seti elde edilmiştir. Veri setine genel bir bakış Şekil 5.3'te sunulmuştur.

5.4. Gensim Kütüphanesi

Gensim (https://radimrehurek.com/gensim), açık kaynaklı bir Python kütüphanesi olup konu modelleme ve doküman benzerliği çalışmalarında kullanılmak üzere geniş kapsamlı yetenekler sunmaktadır.

Bu ve bir sonraki bölümde yer alan çalışmada Gensim'in hazır olarak sunduğu word2vec ve doc2vec algoritmalarının implementasyonları kullanılmıştır. Önerilen yöntemler ve deney senaryoları içerisinde bu algoritmaların nasıl kullanıldığının anlatımı bağlamında Gensim kütüphanesine ait çeşitli özelliklere değinilmiştir. Özellikle doküman modelleme yöntemine ait çeşitli tanım ve parametrelerin açıklanmasının yararlı olacağı düşünülmüştür.

gensim.models.Doc2Vec: Paragraf vektör modeli ya da doküman vektör modeli olarak isimlendirilen, dağıtık doküman temsili modeli eğitiminde kullanılan sınıftır. Cümle, paragraf veya dokümanlar için kullanılabilir. Kelime modeli versiyonu gensim.models.Word2Vec sınıfıdır. Aldığı parametreler: Doc2Vec(vector_size, min_count, dm, epochs, callbacks, workers).

min count: Bir kelimenin (token) derlem (corpus) içerisinde yer alabilmesi için sahip olması gereken minimum frekans değeridir. "Min count" değerinin 1 seçilmesi durumunda (min count = 1) derlemdeki bütün kelimeler sözlükte yer alır.

dm: Bu parametre ile hangi doc2vec algoritmasının kullanılacağı seçilir. Dm=0 olması durumunda distributed bag of words (PV-DBOW), dm=1 olması durumunda distributed memory (PV-DM) algoritması yürütülür. PV-DBOW skip-gram, PV-DM ise CBOW algoritmasına benzerlik göstermektedir.

callbacks: Eğitim sırasında bir dönem (epoch) için başlama veya bitiş anlarında istenen rutinlerin çalıştırılmasını sağlayan parametredir. Bu çalışmada bu parametre aracılığıyla her dönem sonunda modelin performansı test edilmiştir.

workers: Çok iş parçacıklı yürütmede kullanılacak iş parçacığı (thread) sayısını belirler.

gensim.models.doc2vec.TaggedDocument: Gensim kütüphanesinde giriş dokümanı formatındaki nesneye TaggedDocument denir. Bir dokümanı oluşturan kelimelerin listesinden ve o dokümanı belirtici bir veya birden fazla etiketten oluşur.

model.docvecs: Doküman modelinden eğitime katılmış dokümanların temsil vektörlerinin elde edilmesi için kullanılır. Örneğin bir dokümanın etiketi "etiket_X" ise, temsil vektörü şu şekilde elde edilir: *model.docvecs['etiket_X']*. Test verilerinin eğitim verisinde yer almadığı koşulda vektör temsillerinin bu metot ile elde edilemeyeceğine dikkat edilmelidir.

Doc2Vec.infer_vector: Eğitilmiş bir doküman modelinden herhangi bir dokümanın vektör temsilinin elde edilmesinde (infer) kullanılır. Bu metot, eğitim verisinde yer almayan (test verileri) dokümanlar için kullanılabileceği gibi eğitim verisinde yer alan dokümanlar için de kullanılabilir. model.docvecs ile etiketi belli olan, eğitimde yer alan bir doküman için her çağrıda aynı vektör değeri üretilirken, *infer_vector* mekanizmasında her çağrının yeni bir vektör değeri üreteceğine dikkat edilmelidir.

Veri setindeki her bir doküman yani tweet Gensim [134] kütüphanesindeki "Tagged Document" nesnesi yapısına dönüştürülmüştür. Tagged Document (etiketli doküman) nesnesi sözlükte bulunan token birimlerinden o dokümanda bulunanların listesi ('w1', 'w2', 'w3', ... 'wN') ve etiketini (tweet-id) ifade eder:

Tweet[i] = TaggedDocument(['w1', 'w2', 'w3', ... 'wN',], [tweet-id])

Tweet ID'ler Twitter tarafından eşsiz (unique) olarak sunulduğu için her dokümanın ID'si olarak "tweet-id" kullanılmıştır. Bu ID aynı zamanda dokümanın etiketidir. Etiketlemede bir diğer seçenek ise çoklu etiket özelliğinin kullanılmasıdır. Tweet-id ile birlikte kullanıcı-id de doküman etiketi olarak kullanılabilir. Buna benzer ekstra etiket kullanımının doküman modelinin başarısına olan etkileri bu çalışmada incelenmemiştir.

5.5. Veri Ön İşleme Adımları

Farklı metin ön işleme adımları uygulanarak önerilen modelin en başarılı olduğu ön işleme yöntemi araştırılmıştır.

Ön İşleme Yöntemi #1

Veri setinde bulunan bütün tweetler taranarak bir sözlük oluşturulmuştur. Sadece durak kelimeler adı verilen önemsiz token birimleri ve noktalama işaretleri sözlüğe dâhil edilmemiştir. Bunun haricinde bir kez dahi kullanılmış her bir token sözlüğe eklenmiştir. Bu işlemler için Python tabanlı, açık kaynaklı DDİ aracı olan NLTK kullanılmıştır [135]. Ön İşleme Yöntemi #2 ve #3'te bu adımdaki temel temizlik işlemleri aynı şekilde gerçekleştirildikten sonra ekstra adımlar uygulanmıştır.

Ön İşleme Yöntemi #2

URL, *hashtag*, *sayısal değerler* gibi birimlerin özel olarak değerlendirilmesi amacıyla kelime listesine *özel-belirleyiciler* eklenerek bunun yöntemin başarısına etkisi incelenmiştir. Bunun için bütün URL'ler tespit edilip "*http_url*" özel belirteciyle değiştirilmiştir. Benzer şekilde sayısal değerler "*number_value*" ile ve hashtagler de "*hash_tag*" belirteciyle değiştirilmiştir. Şekil 5.4'te orijinal tweetler ve basit özel belirteçlerin uygulanmış halleri gösterilmiştir.

Tweet:	The #cryptocurrency marketcap just crossed $300 billion. Let's go🚀 https://t.co/a6pzsPYfQK
Ön İşleme Yöntemi #2:	*['hash_tag', 'cryptocurrency', 'marketcap', 'crossed', 'number_value', 'billion', 'let', 'http_url']*
Tweet:	Here are the best-dressed celebrities from the 2018 #PCAs: https://t.co/KBIjeoxOgX
Ön İşleme Yöntemi #2:	*['celebrities', 'number_value', 'hash_tag', 'pcas', 'http_url']*

Şekil 5.4. Basit özel belirteç kullanımı örneği

Ön İşleme Yöntemi #3

Diğer bir yöntem olarak özel-belirteçler anlamsal olarak detaylandırılmış ve çoğaltılmıştır. "*per_cent*", "*at_sign*", "*dollar_sign*", "*euro_sign*", "*number_and_word*" gibi belirteçler kullanılmıştır. Şekil 5.5'te orijinal tweetler ve detaylı gelişmiş özel belirteçlerin uygulanmış halleri gösterilmiştir.

Tweet:	$15k @RealTimeWWII $PBL £RDN #MTH $BCPT €GLA %10 https://t.co/oJnQPlzwFq
Ön İşleme Yöntemi #3:	*['dollar_sign', 'number_and_word', 'at_sign', 'realtimewwii', 'dollar_sign', 'pbl', 'pound_sign', 'rdn', 'hash_tag', 'mth', 'dollar_sign', 'bcpt', 'euro_sign', 'gla', 'per_cent', 'number_value', 'http_url']*
Tweet:	Throwback to when 40% of my followers said 75-100%
Ön İşleme Yöntemi #3:	*['throwback', 'number_value', 'per_cent', 'followers', 'said', 'number_and_word', 'per_cent']*

Şekil 5.5. Detaylı özel belirteç kullanımı örneği

5.6. Kullanıcı Gömme Modeli

Veri setinde bulunan her bir tweet birer doküman olarak ele alınarak bir doküman vektör modeli eğitilmiştir. Le and Mikolov'un doküman modelinden [118] yola çıkarak amaç fonksiyonu Denklem 22'deki gibi belirlenmiştir.

$$\frac{1}{L}\sum_{c=1}^{L} \log P\left(w_c|T, w_{c-k}, \ldots, w_{c+k}\right) \quad (5.1)$$

Buradaki amaç fonksiyonu, her bir tweet için, ortalama log-olasılığını maksimize ederek model parametrelerinin ayarlanması şeklinde kurgulanmıştır. Her bir tweetin gömme değeri T ile, tweette bulunan kelimeler $\{w_1, w_2, w_3, \ldots\ w_L\}$, tweet uzunluğu L, bir merkez kelime w_c öncesi ve sonrasındaki diğer kelimelere uzaklık ise k ile gösterilmiştir. Bir kullanıcı için gömme değeri o kullanıcının yazmış olduğu bütün tweetlerin gömme değerlerinin ortalaması alınarak hesaplanmıştır.

$$U = \frac{1}{N}\sum_{j=1}^{N} T_j \quad (5.2)$$

Ayrıca, önceden belirlenen kategoriler için de gömme değerleri hesaplanmıştır. Bu değerler baz alınarak bir kullanıcının kendi grubuna mı yoksa diğer gruplara mı daha yakın olduğu tespit edilebilir. Bu da yöntemin doğru çalışıp çalışmadığını gösterecek göstergelerden birisidir. $\{T_1, T_2, T_3, \ldots\ T_N\}$ vektörleri bir U kullanıcısına ait tweetler olmak üzere; her bir ortalama kategori vektörü şu şekilde hesaplanır:

$$G = \frac{1}{M}\sum_{i=1}^{M} U_i \quad (5.3)$$

$\{U_1, U_2, U_3, \ldots\ U_M\}$ sembolleri G gömme değeri hesaplanan grubun kullanıcılarını göstermektedir.

5.7. Sonuçlar

Önermiş olduğumuz yöntemle elde edilen kullanıcı temsillerinin kalitesi çeşitli senaryolarla test edilmiştir. Manuel olarak hazırlanmış veri setinin altın standart olarak kullanılmasıyla; kullanıcıların grup-temsillerine ve diğer kullanıcı-temsillerine olan benzerlik oranlarına göre yöntemin başarısı görülmüştür.

5.7.1. Genel Başarı Ölçütü

Genel başarı ölçütü olarak (general-accuracy); en çok ait olduğu grup temsiline benzerlik gösteren kullanıcı temsillerini doğru, farklı bir gruba daha çok benzerlik gösteren kullanıcı temsillerini ise hatalı olarak hesaplayarak bir test gerçekleştirilmiştir. Genel başarı ölçütü açısından 20 boyutlu gömmelerin 350 iterasyon sonucundaki değerleri %95.2 olarak en iyi sonucu vermiştir. Bu başarı değeri her grup için ayrı ayrı Tablo 5.1'de verilmiştir.

Tablo 5.1. Grup bazında genel başarı oranları

Grup	Doğruluk oranı (%)
Moda ve yaşam tarzı	100/100
Siyaset	98/100
Teknoloji	97/100
Kripto teknolojisi ve ekonomisi	92/100
Ekonomi ve finans	89/100

Gruplara göre kullanıcı temsillerinin doğruluk oranları incelendiğinde "*moda ve yaşam tarzı*" grubuna ait kullanıcı temsillerinin %100 doğruluk oranıyla doğru tespit edildiği görülmüştür. "*kripto teknolojisi ve ekonomisi*" grubu ile "*ekonomi ve finans*" gruplarının en çok hatalı tahmin yapılan gruplar olduğu görülmektedir. Bunun nedeni bu gruplara ait kullanıcılara ait tweetlerin ayrılabilirliğinin (separability) az ve çoğu zaman iç içe giren konularda olmalarıdır. Veri seti oluşturulurken özellikle birbirine bu derece benzerlik gösteren 2 grup seçilerek yöntemin zorlayıcı şartlar altında göstereceği başarı görülmek istenmiştir. Yöntemin bu yönden başarısının daha kolay incelenebilmesi açısından Tablo 5.2'de grup temsillerinin birbirlerine olan benzerlik oranları verilmiştir.

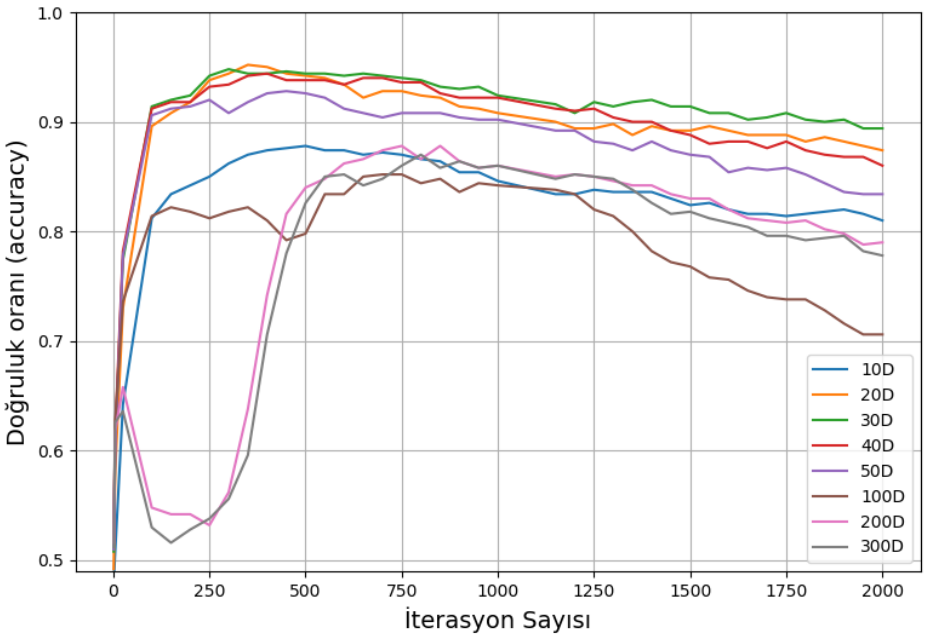
Tablo 5.2. Grup benzerlik matrisi

	Ekonomi	**Kripto**	**Teknoloji**	**Moda**	**Siyaset**
Ekonomi	1	0.822	0.851	0.804	0.804
Kripto		1	0.84	0.75	0.72
Teknoloji			1	0.82	0.78
Moda				1	0.77
Siyaset					1

Yöntemin başarısının farklı parametrelere ve veri ön işleme yöntemlerine göre değişimini gözlemlemek amacıyla da çeşitli deneysel çalışmalar yapılmıştır:

- Veri ön işleme adımlarında sadece önemsiz kelimelerin ve alfa numerik olmayan karakterlerin temizlenmesinin diğer detaylı temizleme ve özel-belirteç kullanımı yöntemlerine göre çok daha az miktarda da olsa daha iyi sonuç verdiği görülmüştür. Bu nedenle genel bir metin ön işleme adımının bu yöntem için en uygun olduğu tespit edilmiştir.

- Modeller en az **2000 iterasyon** boyunca **10, 20, 30, 40, 50, 100, 200 ve 300 boyutlu** vektörler için ayrı ayrı eğitilmiştir. Bu farklı parametrelere göre elde edilen sonuçlar Şekil 5.6'da sunulmuştur.
- Şekil 5.6'da da görüleceği üzere Vektör boyutunun 50 üzerine çıkarılmasının sonuçları kötüleştirmeye başladığı görülmüştür. En iyi başarıyı gösteren **20** ve **50 boyutlu** vektörler en yüksek performansı ilk **500 iterasyonda** vermiştir. Bu durum, doküman modelinin 500 iterasyondan fazla eğitilmesine gerek olmadığını göstermektedir.
- **10, 100, 200 ve 300 boyutlu** vektörlerin en kötü sonucu verdiği görülmüştür. Vektör boyutunun çok düşük veya çok yüksek olarak belirlenmesinin iyi bir tercih olmayacağı ortaya çıkmıştır.



Şekil 5.6. Farklı vektör boyutlarının iterasyon sayısına göre başarısı

5.7.2. Grup Bazında Başarı Ölçütü

Bir diğer başarı ölçüm yöntemi olarak grup gömme başarı değeri önerilmiştir. Bu ölçüm özet olarak bir grup temsili için K-En Yakın Komşu (K-Nearest Nighbor) sınıflandırma algoritmasındaki yaklaşıma benzer bir şekilde en yakın kullanıcılara bakılarak bu kullanıcıların hangi oranda o gruba ait olan kullanıcılar olduğu ölçülmüştür.

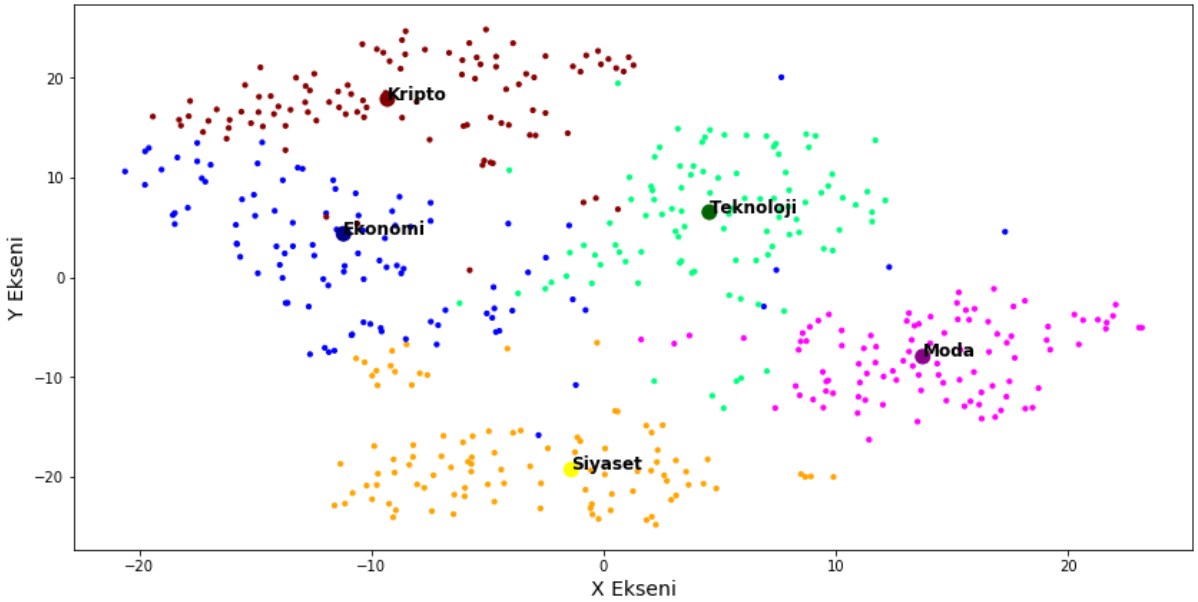
Her gruba en yakın K adet kullanıcı (ilk-K) $\in$ {10, 20, 30, 40, 50, 100} değerleri için ayrı ayrı seçilerek test edilmiştir. En yakın 100 kullanıcının seçilmesi grubun bütün kullanıcılarının test edilmesi anlamına gelmektedir. Grubun toplam kullanıcı sayısından daha az sayıda kullanıcı

seçilmesi durumunda ise grubu en iyi temsil eden kullanıcılar seçilerek temsil başarıları incelenmiştir. Deneylerde en yüksek başarıyı gösteren model (boyut:20, iterasyon:350) için ilk-K başarı oranları Tablo 5.3'te gösterilmiştir.

Tablo 5.3. Grup gömmeleri üzerinden başarı oranları

Grup	ilk-K				
	K=10	K=20	K=50	K=70	K=100
Ekonomi	0.8	0.8	0.6	0.51	0.46
Kripto	1	1	0.94	0.77	0.6
Teknoloji	1	0.85	0.7	0.64	0.58
Moda	1	1	0.98	0.87	0.7
Siyaset	1	0.95	0.74	0.64	0.52

Kullanıcı ve grup temsil vektörlerini görselleştirmek amacıyla t-Distributed Stochastic Neighbor Embedding (t-SNE) algoritması kullanılarak 100 boyutlu gömmeler 2 boyuta indirgenmiştir. Temsil vektörlerinin 2 boyuta indirgenmiş değerleri Şekil 5.7'de gösterilmiştir. Kullanıcılar ait oldukları gruba göre renklendirilmiştir. Büyük noktalar grup temsil vektörlerini göstermektedir. Bu grafikte de görüleceği üzere Ekonomi grubunda olup kendi grubuna en yakın olmayan kullanıcılar çoğunlukla Kripto grubuna yakın çıkmaktadır.



Şekil 5.7. Kullanıcı ve Grup Gömmelerinin 2-Boyutlu gösterimi

5.8. Değerlendirme

Bu çalışmanın sonucunda yorum, retweet, beğeni gibi diğer kullanıcı aktivitesi bilgilerinin de bu yöntemin istifade edebileceği şekilde işlenerek temsil vektörlerinin doğruluk değerini arttırılması fikri ortaya çıkmıştır. Ayrıca veri setinin hazır diğer derlemlerden (Wikipedia, haberler vb.) ve hazır kelime gömmelerinden (word2vec, Glove vb.) faydalanması yönünde araştırmaların yapılmasına karar verilmiştir. Bu çalışmalar Bölüm 6'da sunulmuştur.

6. USER2VEC: SOSYAL MEDYA KULLANICI TEMSİLİ İÇİN YENİ BİR YAKLAŞIM

Tezin bu bölümünde bir önceki bölümde önerilen user2vec yönteminin hem uygulanan temsil modeli hem de kullanılan veri seti bakımından geliştirilmesi yönünde ortaya çıkan çeşitli fikirler ele alınmıştır. Gelişmiş (state of the art) metin temsil yöntemlerinin kullanıcı temsillerinin elde edilmesi amacıyla kullanımı bakımından literatürdeki en kapsamlı çalışmalardan biri sunulmuştur.

Bu çalışmada metin temsil yöntemlerinin sosyal medya kullanıcılarının (Twitter) temsil edilmesinde kullanımı kapsamlı bir şekilde araştırılmıştır. Dağıtık doküman temsillerinin sosyal medya kullanıcılarının temsilinde kullanıldığı genel yaklaşımların ele alındığı önceki çalışmanın sonucunda merak uyandıran yeni soruların cevabı bu çalışmada aranmış ve çeşitli çözüm önerileri geliştirilmiştir.

Kullanıcıları yazdıklarıyla temsil etmek konusunda çeşitli yaklaşımlar denenmiş ve doc2vec tabanlı bir temsil yöntemi önerilmiştir. BERT, ELMO, klasik word2vec gibi güncel kelime temsil yaklaşımları; Glove, Numberbatch, FastText gibi önceden eğitilmiş kelime temsil modelleri ve doc2vec yaklaşımının sosyal medya kullanıcılarının yazdıklarıyla temsil edilmesindeki başarıları incelenmiştir. Ayrıca beğeni, retweet, yorum, konum gibi metinsel olmayan özelliklerin metin içerisine eklenerek eğitilen doküman temsil modelleriyle gösterdiği başarı incelenmiştir.

Bu bölümde gerçekleştirilen çalışmalar "User Representation Learning for Social Networks: An Empirical Study" isimli makale ile yayınlanmıştır [136].

6.1. Problemin Tanımı ve Amaç

Bu çalışma önceki user2vec çalışmasının devamı niteliğindedir. Önceki çalışmada basit olarak dağıtık doküman temsil yöntemiyle vektör temsilleri oluşturulacak kullanıcıların tweetleri birer doküman olarak ele alınarak eğitilen doc2vec modelinden elde edilen gömmeler her bir kullanıcı için kendi yazdığı tweetlerin gömmelerinin ortalamasının alınması suretiyle elde edilmiştir. Yöntemin başarısı altın standart bir veri seti ile incelenmiştir ve modellerin başarıları farklı ön işleme yöntemleri bakımından karşılaştırılmıştır. Ayrıca, farklı vektör boyutları ayrı ayrı denenerek parametre seçiminde vektör boyutunun ve iterasyon sayısının etkisi incelenmiştir. Model eğitiminde derlem olarak sadece temsili elde edilecek olan kullanıcıların tweetleri kullanılmıştır. Bu yöntemin geliştirilmesi ve gerçek dünyada uygulanması bakımından çeşitler hususlar ortaya çıkmıştır. Bu hususlarla ilgili olarak aşağıda belirtilen katkılar sunulmuştur:

i. Gensim kütüphanesi örneği üzerinden ele alınırsa; doküman_modeli(['doküman_etiketi']) şeklinde bir fonksiyon çağırarak model eğitiminde kullanılan derlemde yer almayan tweetlerin temsilleri elde edilemez. Çünkü model eğitimi sırasında her bir dokümana karşılık gelen vektör uzayı değerleri bir amaç fonksiyonuna göre sadece derlemde yer alan diğer dokümanlarla göre belirlenmektedir. Bunun yerine ancak infer_vector fonksiyonu ile modelin daha önce karşılaşmadığı dokümanlar için vektör değeri ürettiği inferring (çıkarım) mekanizması kullanılabilir. Fakat bu mekanizma işletildiğinde dikkat edilmesi gereken önemli bir husus ortaya çıkmaktadır. Bu üretken yöntem tamamen deterministik değildir ve aynı metne yönelik sonraki çağrılar vektör uzayında birbirine yakın konumlanan değerlere sahip olsa da birbirinden farklı vektörler üretir. Bu çalışmada modellerin performansları değerlendirilirken eğitim verisi içerisine test kullanıcılarına ait tweet, yorum, takipçi gibi herhangi bir bilgi dahil edilmemiştir. Bu şekilde modelin başarısı gerçek hayatta uygulama senaryosunda göstereceği performans üzerinden değerlendirilmiştir.

ii. Manuel olarak seçilmiş kullanıcılardan oluşan altın standart bir eğitim veri setinin mevcut olması her zaman mümkün değildir. Bu nedenle alternatif derlemlerin nasıl başarı göstereceği önemli bir sorudur. Doküman modelinin eğitiminde farklı veri setlerinin kullanılmasının kullanıcı temsillerinin başarısına olan etkisi incelenmiştir. Hazır olarak kullanılabilen (public dataset) Twitter, Wikipedia gibi kaynaklardan elde edilmiş veri setlerinin altın standart veri setinin yerine kullanılması durumunda nasıl başarı gösterdiği incelenmiştir. Ayrıca bu veri setlerinin farklı kombinasyonlarda bir araya getirilmesiyle elde edilen karışık veri setlerinin başarısı incelenmiştir.

iii. Doc2vec modellerinin hem PV-DM hem de PV-DBOW algoritmalarıyla; min_count 1 ve min_count 5 değerleri için ayrı ayrı denenerek parametre araştırması gerçekleştirilmiştir. Bu parametrelerin açıklamaları Bölüm 5.4'te yapılmıştır.

iv. Doküman vektörlerinin elde edilmesinde; geleneksel bir yöntem olan TF-IDF'in; Glove, FastText gibi PWE'lerin ve BERT, ELMO gibi cümle kodlayıcılarının (encoder) kullanımının çeşitli yönleriyle incelenmesi yapılmıştır.

Önerilen yöntemlerin karşılaştırılması rastgele seçilmiş olan bir test kümesinde yer alan Twitter kullanıcılarının temsil vektörlerinin doğru elde edilmesindeki başarı üzerinden yapılmıştır. Kullanıcı temsillerinin doğruluğu kişileri olması gereken gruplarda temsil etmesi ve benzer kullanıcıları vektör uzayında birbirine yakın olarak konumlandırması konusunda gösterdiği başarı ile tespit edilmiştir. Kesinlik, duyarlılık (hassasiyet), doğruluk ve F1 skorları verilmiştir.

6.2. Kullanılan Metin Temsil Yöntemleri

Sosyal medya analizi konusunda yapılan araştırmaların başlıca zorluklarından biri; karmaşık, yapısal olmayan (unstructured) metin ve sosyal ağ verileridir. Bu verilerin analizinde DDİ yöntemlerinden yararlanılarak geliştirilmiş yeni yöntemler ortaya çıkmaktadır. Gelişmiş kelime ve doküman temsil yöntemlerinin ayrıca bilgi erişimi, tavsiye sistemleri, güvenlik ve daha birçok alanda ilerlemelere katkı sağladığı görülmektedir [137]. Bu çalışmada kapsamlı bir şekilde farklı metin temsil yöntemleri kullanılmıştır. Kullanılan yöntemlerin listesi Tablo 6.1'de verilmiştir.

Tablo 6.1. Kullanılan metin temsil yöntemleri

Yöntem türü	Yöntem/Yöntemler
Eski yöntemler	TF-IDF
Klasik metin temsil yöntemleri	Word2vec, doc2vec
Önceden eğitilmiş kelime temsilleri (PWE)	FastText, Glove, Numberbatch, Google News
Bağlamsal (contextual) metin temsilleri	Bert, Elmo, Stacked Embedding

Kullanılan metin temsil yöntemlerine ait açıklamalar ve ilgili literatür bilgisi Bölüm 2'de verildiği için bu bölümde yeniden anlatılmamıştır.

6.3. Veri Setleri

Bu çalışmada, Twitter kullanıcılarının temsil edilmesi görevinde farklı yöntemler kullanılmıştır. Ayrıca, yöntemlerin tür ve kaynak bakımından farklı eğitim verilerine göre gösterdiği performansın incelenmesi amacıyla detaylı bir karşılaştırma yapılmıştır. Bu bölümde veri setlerinin yapısı ve elde edilme süreçleriyle ilgili bilgiler yer almaktadır.

Önceki çalışmada kullanılan ekonomi, kripto-ekonomi, teknoloji, siyaset ve moda alanlarının her biri için belirlenmiş 100'er adet Twitter kullanıcısının sadece kendi yazdıkları tweetlerin bulunduğu veri seti bu çalışmada kullanılan ana veri setidir. 500 adet Twitter kullanıcısının her birinin en güncel 1000 tweetinden meydana gelmek üzere toplam 500 bin tweet bulunmaktadır. Bu çalışmada bu veri setindeki kullanıcılar sırasıyla %60'a %40 oranında eğitim ve test kullanıcıları olmak üzere bölünmüştür. Eğitim seti olarak ayrılan bölümde yer alan 300 kullanıcıya ait 300.000 tweet **300K-TW** olarak anılacaktır. Bu veri seti Twitter REST API ile elde edilmiştir.

300K-TW veri setinin ek bilgilerle zenginleştirilmesiyle yeni veri setleri oluşturulmuştur. Twitter REST API ile her bir kullanıcının profil açıklaması ve konum bilgileri toplanmıştır. Bu bilgilerden oluşan veri kümeleri bir araya getirilerek "*Name"* (kullanıcı ismi), "*Location"*

(konum), "*Description"* (açıklama) kelimelerinin ilk harfleriyle **NLD** isimli bir veri kümesi elde edilmiştir. Konum ve profil açıklamaları her kullanıcı için mevcut olmayan, kullanıcı tercihine göre paylaşılan bilgiler olduğu için NLD veri kümesinde bazı kullanıcılar için ilgili bilgilerin boş olma ihtimalleri göz önünde bulundurulmalıdır. İsim-konum ve isim-açıklama veri kümelerine genel bir bakış Şekil 6.1 ve Şekil 6.2'de sunulmuştur. Tweetlerin kullanıcı ismi ve konum bilgisiyle bir araya getirildiği ilgili bir çalışma Xing ve arkadaşları tarafından gerçekleştirilmiştir [132]. Bu çalışmada da kullanıcı ve konum bilgilerinden oluşan benzer bir veri seti (**NL**) oluşturularak karşılaştırma yapma imkânı sağlanmıştır.

1	**Ba[illegible]**	**Washington, DC**
2	ch [illegible] ld	Washington, DC
3	Aco [illegible]	Washington, DC
4	mic[illegible] malkin	USA
5	S[illegible] yan	Janesville, WI
6	[illegible] hamAn [illegible] le	DC
7	AO[illegible]	Bronx + Queens, NYC
8	TH[illegible]	Stockbridge, Georgia
9	Sa[illegible]	London, England
10	B[illegible] nson	United Kingdom

Şekil 6.1. İsim-konum (NL) verilerine genel bakış

1	**Barack[illegible]ma**	**Dad, husband, President, citizen.**
2	[illegible]	Moderator of @[illegible]press and @[illegible]ews political director; C[illegible]
3	Ac[illegible]cta	CNN's Chief White House Correspondent. I believe in #realnews.
4	mic[illegible]malkin	Little brown woman. Big mouth. Wife. Mom. Entrepreneur. Geek. [illegible]
5	Spea[illegible]yan	Office of the [illegible]th Speaker of the House, F[illegible]an.
6	Ing[illegible]gle	Mom, author, The [illegible] Ar[illegible]e 10p ET Fox News, podcast on [illegible]
7	AO[illegible]	Congresswoman for NY-14 (the Bronx & Queens). In a modern, m
8	TH[illegible]in	Official Twitter for [illegible]n. Tell YOU the Truth. Listen 24/7 h
9	Sadiq[illegible]	Mayor of London. #Team[illegible] #DeliveringForLondon
10	Bori[illegible]	Prime Minister of the United Kingdom and @Conservatives leade[illegible]
11	mark[illegible]	CBS News White House Correspondent

Şekil 6.2. İsim-açıklama veri kümesine genel bakış

Bu çalışmanın asıl amaçlarından biri, kullanıcı hakkında elde edilen bütün bilgilerin bir arada kullanımına yönelik bir çözüm geliştirmektir. Bu nedenle NLD veri kümesindeki bilgiler ile kullanıcı aktivite verileri bir araya getirilerek **NLD-RLC** isimli ayrı bir veri seti oluşturulmuştur. "*Retweet*", "*Like*" ve "*Comment*" kelimelerinin ilk harfleriyle isimlendirilmiş **RLC** veri kümesi; bir kullanıcının tweetleriyle herhangi bir şekilde etkileşimde bulunmuş kullanıcıların kullanıcı-adlarından oluşmaktadır. Örneğin, Bill Gates kullanıcısının 12131415 id'li tweetini paylaşan (retweet), bu tweeti beğenen (like) ve bu tweet altına yorum (comment) yazan kişilerin kullanıcı isimleri o tweet için RLC kümesini oluşturmaktadır. Bu şekilde bir kullanıcının içerisinde yer aldığı sosyal ağ içerisinden yapısal olmayan bilgiler elde edilmiştir. İçeriği kullanıcı adlarından oluşan bu veri seti tek başına kullanılmak için değil, tweetlerden oluşan 300K-TW veri setinin sosyal ağ bilgileriyle zenginleştirilmesi amacıyla oluşturulmuştur. Benzer şekilde normalde coğrafi bir bilgi olan konum bilgisi de kullanıcının tanınmasında etkisi olabilecek ilave bir bilgidir ve tweet veri setinin zenginleştirilmesinde kullanılmıştır. Tablo 6.2'de bu çalışmada kullanılan eğitim veri setleri açıklamalarıyla birlikte listelenmiştir.

Tablo 6.2. Kullanıcı temsili eğitim verisi listesi

İsim	ID	İçerik	Sayılar
300K-TW	A	Önceden belirlenen 5 kategori için elle seçilmiş kullanıcıların tweetleri. Önceki çalışmada kullanılan veri setidir.	Her kategoriden 60'ar kullanıcının son 1000 tweeti; 3**00 bin tweet.**
1M-Wiki	B	20 Haziran 2019 tarihli, 4.672.834 adet doküman içeren, İngilizce Wikipedia dökümü (dump).	50 ila 1000 karakter uzunluğunda, rastgele seçilen **1 Milyon doküman.**
1M-TW	C	Harward News Outlet [138] veri setindeki tweet id'lerden elde edilen tweet metinleri.	İndirilip dehidre edilen 39.695.156 tweet içerisinden rastgele seçilen **1 Milyon tweet.**
NL	D	300K-TW veri setindeki her bir kullanıcının kullanıcı adı ve konum bilgileri.	**240 kullanıcının** bilgileri**.**
NLD-RLC	E	300K-TW tweetlerinin kullanıcı adı, açıklama, konum bilgileri ile retweet, like, comment eden hesapların kullanıcı isimleri.	**300 bin tweet** ve **240 kullanıcıya** ait bilgiler.

Tezin devamında veri setleri kısa isimler olarak verilen A, B, C, D, E id'leriyle anılacaktır. Tekrar vurgulamak açısından; D ve E veri setlerinin tweet metinleri içermeyen ek

bilgilerden oluştuğuna ve A veri setinden ilgili tweetlerle bir araya getirilerek kullanıldığına dikkat edilmelidir. Yani; A+D, A+E, A+D+B, A+E+B gibi kombinasyonlarla bir araya getirilerek kullanılmıştır.

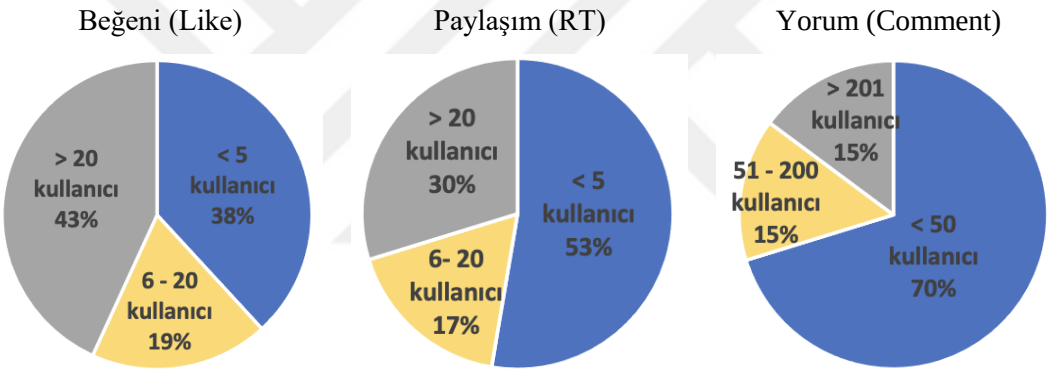
Büyük miktarda tweetin herkese açık, indirilebilir şekilde paylaşılması Twitter kurallarına aykırı olduğu için Harward News Outlet veri setinde tweetler dehidre edilmiş olarak sadece tweet id'leriyle sunulmaktadır. Bu nedenle C veri seti oluşturulurken tweet id'lerinden tweet metinlerinin elde edilmesi (rehidrasyon) sağlanmıştır. Bu işlem için bir tweet toplama aracı olan Python tabanlı Twarch [139] yazılımı kullanılmıştır. C veri setinde televizyon kanalları ve gazeteler gibi büyük medya kuruluşlarına ait Twitter hesaplarından toplanılmış tweetler bulunmaktadır.

Bildiğimiz kadarıyla, semantik gruplara ayrılmış sosyal medya kullanıcılara ait etkileşim verilerinin toplanması daha önce denenmemiş bir yaklaşımdır. Standart Twitter API ve bu API üzerinden Twitter verilerine erişim yetenekleri sağlayan Tweepy [140], Twarc, Twitter4J [141], Bear [142] gibi kütüphaneler (wrapper-library) bir kişinin paylaştığı, beğendiği veya yorum yazdığı tweet bilgilerinin elde edilmesine API kullanım limitleri dâhilinde olanak vermektedir. Fakat bir tweeti kimlerin beğendiğinin ya da paylaştığının listesini bu ücretsiz ve açık kaynaklı olan API'ler ile elde etmek mümkün değildir. Bunun için bazı dolaylı yollar mevcut olsa da binlerce tweet için yeterli performansı göstermemektedir. Çünkü elde edilecek sonuçlar tarih, kullanıcı popülerliği, tweetin ulaştığı kullanıcı sayısı gibi farklı parametrelerden etkilenerek yetersiz bir sonuç vermektedir. Bununla birlikte normal bir kullanıcının bir web tarayıcı üzerinden Twitter web sayfasına yapacağı gezintide bile bir tweeti paylaşan, beğenen kullanıcıların ("liked by", "retweeted by" listeleri) ancak sınırlı bir kısmı görüntülenebilir. Bu listeler genellikle en fazla 100 kullanıcı bulunacak şekilde sunulmaktadır. Yani bir tweeti beğenen kullanıcı sayısı 1.000 de olsa 100.000.000 de olsa en fazla bu kullanıcılardan 100 tanesinin kullanıcı adı görüntülenebilir.

RLC veri kümesinin elde edilmesinde Selenium (https://www.selenium.dev/) ve BeautifulSoup [143] kütüphaneleri kullanılarak 500.000 tweetin sırayla tarayıcıda otomatik olarak görüntülenerek (crawling) tweeti paylaşan, beğenen ve yorum yazan kullanıcı listeleri otomatik olarak kaydırılmak (autoscroll) suretiyle kazınmıştır (scraping). Bu işlem, bir adet kullanıcının 1000 tane tweeti için ortalama 4 ila 8 saat sürmektedir. Geliştirilen veri toplama programı bir bilgisayarda 7/24 çalışacak şekilde ayarlanmıştır. Bütün kullanıcılar için veri kazıma adımlarının tamamlanması yaklaşık 4 ay sürmüştür. Bu veri setine ait bazı önemli bilgiler şunlardır:

- Paylaşım, beğeni ve yorum sayıları tweetler arasında sabit değildir ve çevrimiçi olarak değişmektedir. Geliştirilen veri kazıma programı anlık olarak görüntülenen kullanıcı isimlerini toplamaktadır.

- Yaklaşık 4 ay süren veri kazıma sürecinin tamamlanmasıyla 3.599.588 farklı kullanıcı adı toplanmıştır. Bu verilerin paylaşım, beğeni ve yorum aktivitesi bakımından dağılımları Şekil 6.3'te verilmiştir. Grafik incelendiğinde; tweetlerin %43'ünden en az 20'şer adet beğeni etkileşiminde bulunan kullanıcı adı elde edildiği; tweetlerin %17'sinde RT etkileşiminde bulunan 6 ile 20 arasında sayılarda kullanıcı adı elde edildiği gibi detaylı bilgiler görülmektedir.
- "Moda" başlığı altında yer alan 100 kullanıcının 100.000 adet tweeti ile beğeni, paylaşım ve yorum yazma sonucunda etkileşimde bulunmasından dolayı 803.920 adet kullanıcı adı elde edilmiştir. RLC veri kümesinde yer alan kullanıcı isimlerinin grup bazında dağılımlarının tamamı Tablo 6.3'te verilmiştir. Tabloda ayrıca yazım kolaylığı sağlamak açısından gruplara verilen kısa isimler (ID) yer almaktadır.



Şekil 6.3. Beğeni, paylaşım ve yorum aktiviteleri veri dağılımı

E veri seti tweetlerle birlikte ilgili metinsel olmayan (non-textual) özelliklerin bir arada toplandığı karmaşık bir veri setidir. Veri setinde yer alan tweetlerle *paylaşma, beğeni* ve *yorum* aktivelerinde bulunan, yüzbinlerce kullanıcı adlarından oluşan listeler **kullanıcı aktivite verisi** olarak adlandırılabilir. Bu listelerin farklı kullanıcı adlarından oluşan kümeler olarak ele alınması ve elde edildikleri G1, G2, G3, G4, G5 grup ID'leriyle gösterilmesi durumunda:

$$G1 \cap G2 \cap G3 \cap G4 \cap G5 = 1171$$

1171 adet kullanıcı adı 5 grubun her birinde en az 1 defa yer almaktadır. Bu kesişim değerinin yanı sıra, kullanıcı aktivitelerinden oluşan veri setinin özelliklerinin daha detaylı olarak ortaya konulması maksadıyla gruplar arasındaki detaylı kesişim ve benzerlik değerleri incelenmiştir. Kullanıcı listeleri bakımından grup benzerliklerinin tespit edilmesi için kabaca gruplar arasındaki ortak kullanıcı adı sayısına bakılabilir. Tablo 6.4'te görüleceği üzere "Ekonomi" ve "Politika" grupları bu bakımdan en benzer gruplardır. Bu gruplar aynı zamanda

veri seti içerisinde en fazla ve en az sayıda kullanıcı adı içeren kullanıcı aktive listeleridir. Dolayısıyla, bu basit yaklaşımın, gerçek anlamda bir bezerlik ölçemekten ziyade eleman sayısı en fazla olan kümenin benzerlik karşılaştırmasında daha öne geçmesi yönünde bir sonuç verme ihtimali bulunmaktadır. Ayrık eleman sayılarının dengesiz olduğu bu tarz koşullarda Jaccard [144] ve Örtüşme katsayısı (Overlap coefficient veya Szymkiewicz–Simpson coefficient) [145] gibi yöntemlerle küme benzerlikleri daha sağlıklı bir şekilde ölçülebilir.

Tablo 6.3. Gruplara göre kullanıcı sayısı dağılımı

Grup ID	Grup Adı	Kullanıcı-adı sayısı
G1	Ekonomi	247.109
G2	Kripto ekonomi	340.653
G3	Teknoloji	649.865
G4	Moda	803.920
G5	Politika	1.278.024

Tablo 6.4. Kullanıcı aktivite verilerinin gruplar arası kesişim sayıları

∩	G1	G2	G3	G4	G5
G1	450162	57347	65247	15561	**122840**
G2		439652	44633	6782	25580
G3			817242	29839	86233
G4				887604	57264
G5					1515083

Gx ve Gy kullanıcı adı listelerinin kesişimlerinin birleşimlerine oranı $|Gx \cap Gy|/|Gx \cup Gy|$ ile Jaccard benzerlikleri bulunur. Gruplarda bulunan farklı eleman sayıları ve gruplar arasındaki kesişim değerleri bilindiğinden $|Gx \cup Gy| = |Gx| + |Gy| - |Gx \cap Gy|$ ile birleşim değerleri hesaplanabilir. Gruplar arsındaki Jaccard benzerlik değerleri Tablo 6.5'te verilmiştir.

Tablo 6.5. Kullanıcı listelerinin gruplar arası Jaccard benzerlikleri

Jaccard Benzerliği	G1	G2	G3	G4	G5
G1	1	**0.069**	0.054	0.012	0.067
G2		1	0.037	0.005	0.013
G3			1	0.018	0.038
G4				1	0.024
G5					1

Örtüşme katsayısı benzerliklerinin bulunması için ise kümelerin kesişim sayılarının küçük elemanlı olan kümenin eleman sayısına olan oranı $|Gx \cap Gy|/\min(|Gx|,|Gy|)$ ile hesaplanır. Örtüşme katsayısı benzerlikleri Tablo 6.6'da verilmiştir.

Tablo 6.6. Kullanıcı listelerinin gruplar arası örtüşme katsayısı benzerlikleri

Örtüşme Katsayısı	G1	G2	G3	G4	G5
G1	1	0.130	0.145	0.035	**0.273**
G2		1	0.102	0.015	0.058
G3			1	0.037	0.106
G4				1	0.065
G5					1

Bu çalışmadaki en kapsamlı veri seti A ile E veri setlerinin birleştirilmesiyle elde edilmiştir. Tweet yazarının açıklama ve konum bilgileriyle o tweeti beğenen, paylaşan veya yorumlayan kişilerin kullanıcı isimleri bir araya getirilmiştir. Bu yaklaşım metinsel özelliklerle metinsel olmayan özelliklerin bir nevi kaynaştırılmasıdır. Birleştirme öncesinde, Xing ve Paul'un [132] da dikkat çektiği, sahte birim (dummy token) kullanımının gerekliliği göz önünde bulundurularak öznitelikler bir dolgulama (padding) işleminden geçirilmiştir. Dolgulama sayesinde metinsel olmayan özniteliklerin metin içerisinde geçen kelimelerle doğrudan aynı pencere içerisinde yer almaları engellenmiştir. Bunun için önerilen yaklaşıma ait adımlar şu şekildedir:

- Dummy token olarak "***enrtag***", "***nametag***", "***loctag***", "***desctag***", "***liketag***", "***rttag***", "***commtag***" kelimeleri kullanılmıştır. Bu birimler aynı pencerede yer aldıkları özniteliklerin yanında kukla kelime görevi görmektedir.
- Tweet ve zenginleştirme verileri (konum, açıklama, beğeni vb.) birleştirilirken önce "*enrtag*" ile zenginleştirme işleminin başladığı bilgisi kodlanmıştır. Bu çalışmada kullanılan doküman modellerindeki maksimum pencere boyutu 5 olduğu için "*enrtag*" ve diğer bütün birimler, ilgili öznitelik öncesinde ve sonrasında 5'er kez yan yana yazılarak eklenmiştir.
- "A + E" veri setinin her bir satırında yer alan tweetin yazarının kullanıcı adı, öncesinde ve sonrasında "*nametag*" birimleri ile dolgulanarak eklenmiştir. Aynı yöntemle; konum bilgisi için "*loctag*", açıklama için "*desctag*", beğenide bulunan her bir kullanıcı adı için "*commtag*", paylaşan her bir kullanıcı adı için "*rttag*" ve o tweetin altına yorum yazan her bir kullanıcının kullanıcı adı için "*commtag*" yazılarak birleştirme işlemi gerçekleştirilmiştir.
- Bu adımların gerçekleştirildiği örnek bir kayıt Bölüm 6.4.2'de yer almaktadır.

6.4. Önerilen Yöntemler ve Deney Senaryoları

Farklı metin temsil yöntemlerinin kullanıcı gömmesi elde edilmesinde kullanımı kapsamlı olarak araştırılmıştır. Modellerin eğitiminde A, B, C, D ve E veri setleri (Tablo 6.2) kullanılmıştır. Veri setlerinin farklı senaryolar için bir arada kullanımı noktasında model performansları ayrıca incelenmiştir. Model giriş verisi bir kullanıcının tweetleri, aktivite verisi, profil bilgileri veya bunların farklı kombinasyonlarda birleşimidir. Model çıktısı ise sabit uzunluklu bir $\mathbb{R}^d$ vektörüdür.

Modellerin asıl eğitim ve test verileri sosyal medya kullanıcılarının kendileridir. Modellerin eğitiminde bu kullanıcıların bilgileri (tweet, aktivite, profil bilgileri) kullanılmaktadır. Yani, $\boldsymbol{S}$ tane önceden belirlenen gruba ait $\boldsymbol{M}$ adet kullanıcının bilgileriyle modeller eğitilmiştir. Aynı şekilde, önceden belirlenen $\boldsymbol{S}$ gruba ait $\boldsymbol{N}$ test kullanıcısının bilgileri ile modellerin başarısı ölçülmüştür. Başarı ölçütü olarak test verisinde yer alan kullanıcı vektörlerinin eğitim verisinde yer alan kullanıcılara olan benzerliği kullanılmıştır. Bir kullanıcıyı temsil eden vektörün, yani kullanıcı gömme değerinin doğru olarak kabul edilmesi için eğitim veri seti içerisinde ait olduğu grubun temsiline diğer gruplara göre daha yakın olması gözetilmiştir. Grup temsil değeri; bir gruba ait kullanıcıların ortalama vektör temsil değeri olarak belirlenmiştir.

Bütün yaklaşımlarda eğitim verisinde bulunan her bir tweet için d-boyutlu bir vektör $\vec{T} \in \mathbb{R}^d$ üretilir. Bu aşama her yaklaşımda farklı bir metin temsil yöntemiyle (*Temsil_Modeli*) gerçekleştirilmiştir. Kullanıcı temsilleri elde edilirken tweet vektörlerinin eleman bazında (element-wise) ortalaması alınır:

$$\frac{1}{K}\sum_{v=1}^{K} Temsil\_Modeli(T_v) \tag{6.1}$$

Burada $\boldsymbol{K}$, bir kullanıcının tweet sayısını göstermektedir. Her bir grup temsili $\boldsymbol{\theta_k}$ ($k = 1, \dots, S$) grubun eğitim verisinde yer alan kullanıcı temsillerinin $\boldsymbol{\alpha_i}$ ($i = 1, \dots, M$) eleman bazında ortalamasıyla hesaplanır. Test kümesinde yer alan kullanıcı temsillerinin $\boldsymbol{\beta_j}$ ($j = 1, \dots, N$) her biri için kullanıcının ait olması gereken grup temsiline olan uzaklığı ile diğer gruplara olan uzaklıkları bulunarak karşılaştırılır. Kullanıcı ve grup gömmeleri üzerinde gerçekleştirilen uzaklık ölçümleri için Öklid uzaklığı tercih edilmiştir [146].

$\{\mathbb{E}_{\boldsymbol{\alpha}}^{\boldsymbol{i}}, \mathbb{E}_{\boldsymbol{\beta}}^{\boldsymbol{j}}, \mathbb{E}_{\boldsymbol{\theta}}^{\boldsymbol{k}} \in \mathbb{R}^d\}$ d-boyutlu gömme (embedding) vektörleri olmak üzere; $\mathbb{E}_{\alpha}^{i}$ eğitim kümesinde yer alan α_i kullanıcısının, $\mathbb{E}_{\beta}^{j}$ test kümesinde yer alan β_j kullanıcısının temsilini göstermektedir. θ_k grubunun ortalama grup temsili $\mathbb{E}_{\theta}^{k}$ o grupta yer elan eğitim kullanıcılarının gömmeleriyle şu şekilde hesaplanır:

$$\mathbb{E}_{\theta}^{k} = \frac{S}{M}\sum_{i=1}^{M} \mathbb{E}_{\alpha}^{i} \ , \ \alpha_i \in \theta_k \tag{6.2}$$

j. test kullanıcısı ve k. grubun temsillerinin Öklid uzaklığı; $d_{jk} = \left|\mathbb{E}_{\beta}^{j} - \mathbb{E}_{\theta}^{k}\right|$ ile gösterilmiştir. Şekil 6.4'te modelin ürettiği (infer) temsillerin doğruluk testi için gerçekleştirilen adımlar verilmiştir. Modelin genel başarısı, temsili doğru tahmin edilen kullanıcı sayısı (true positive - TP) üzerinden hesaplanır.

Denklem 6.1 ile test kullanıcılarının temsillerini bul; β_j ($j = 1, \dots, N$)
Denklem 6.2 ile ortalama grup temsillerini bul; θ_γ ($\gamma = 1, \dots, S$)
for *her bir test kullanıcısı* β_j
    kullanıcının ait olduğu grubu belirle; $\beta_j \in \theta_k$
    $d_{jk} = \left|\mathbb{E}_{\beta}^{j} - \mathbb{E}_{\theta}^{k}\right|$ ile kendi grubuna uzaklığını hesapla
    sonuc = Doğru
    for *her bir grup* θ_γ
        if $\beta_j \notin \theta_\gamma$ *// kullanıcının kendi grubu değilse*
            $d_{j\gamma} = \left|\mathbb{E}_{\beta}^{j} - \mathbb{E}_{\theta}^{\gamma}\right|$ ile sıradaki gruba olan uzaklığını hesapla
            if $d_{jk} > d_{j\gamma}$ *// diğer gruplardan birine daha yakınsa*
                sonuc = Yanlış. //kullanıcı temsili hatalı
    end for
    if *sonuc == Doğru. // kendi grubuna en yakınsa*
        doğru tahmin sayısını 1 arttır
end for
genel başarı (accuracy) = doğru tahmin sayısı / N

Şekil 6.4. Kullanıcı temsili genel başarı hesaplama adımları

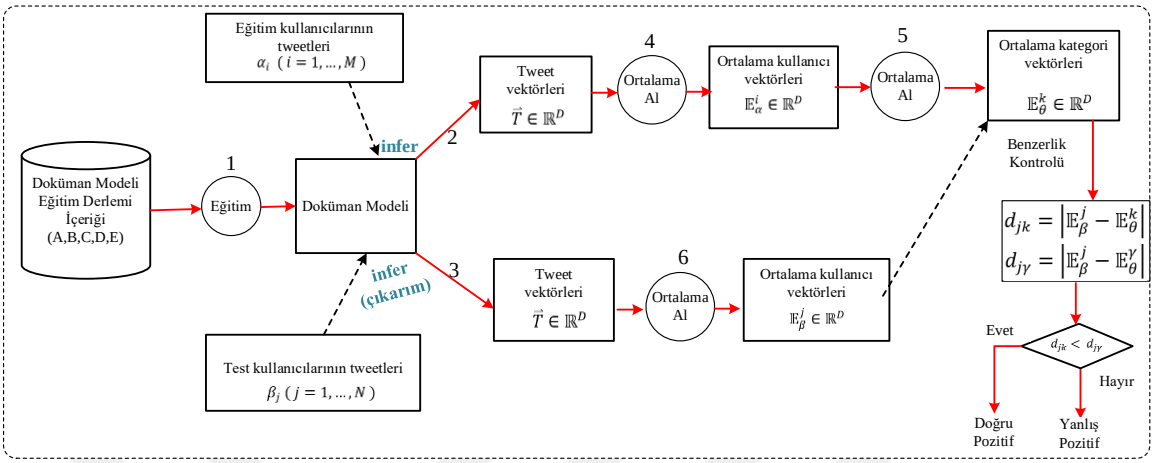
Önerilen kullanıcı temsili yönteminin gerçek hayatta uygulanması durumunda önceden belirlenmiş kategorilerin manuel olarak seçilmiş kullanıcılarından oluşturulan bir veri setine ihtiyaç duyulmamaktadır. Model, eğitim aşamasında grup etiketlerini dikkate almamaktadır ve tamamen denetimsiz bir öğrenme gerçekleştirmektedir. Fakat kullanıcı temsillerinin doğruluğunu test etmek için ortalama grup temsilleri kullanılmıştır. Önerilen yöntemin kapsamlı deneyleri çeşitli veri setleri ve farklı metin temsil teknikleri uygulanarak test edilmiştir. Bu yaklaşımlar üç ana bölümde incelenmiştir. Bunlar;

- Doc2vec yaklaşımı,
- Doc2vec yaklaşımının kullanıcı verileriyle zenginleştirilmesi,
- Kelime gömme yöntemlerinin uygulanmasıdır.

6.4.1. Doc2vec Yaklaşımı

Bu yaklaşımda eğitim veri seti olarak belirlenen bir derlem ile bir doc2vec modeli eğitilir. Eğitilen model ile eğitim ve test kullanıcılarının temsilleri elde edilir. Bir önceki çalışmada önerilen dağıtık doküman temsili modeli ile aynı amaç fonksiyonu (Denklem 5.1) kullanılmıştır.

Model eğitimi, temsil vektörlerinin elde edilmesi ve Şekil 6.4'te verilen genel başarı ölçme adımlarını kapsayan aşamaların bütünü Şekil 6.5'te özetlenmiştir.



Şekil 6.5. Doc2vec modeli ile kullanıcı temsil mimarisi

Şekil 6.5'te yer alan 1, 2 ve 3 numaralı adımlar doc2vec yaklaşımı için metin temsillerinin elde edilmesi aşamalarını göstermektedir ve önerilen yaklaşımlar arasında farklılık göstermektedir. Diğer adımlar (4, 5 ve 6) ise bütün yaklaşımlarda aynı şekilde uygulanmıştır.

Tablo 6.7. Doc2vec yaklaşımı (Deney senaryoları #1)

No.	Veri seti	Senaryo açıklaması
1	A	Problemde ele alınan kullanıcılar göz önünde bulundurularak elle seçilmiş sınırlı miktarda kullanıcıya ait tweetin model eğitiminde kullanılması.
2	B	Eğitim aşamasında hiç tweet kullanılmaması. Büyük boyutlu, genel amaçlı, hazır derlemlerin kullanılması. Örneğin Wikipedia, haberler vb.
3	C	Rastgele seçilmiş yüksek miktarda kullanıcıya ait çok sayıda tweetin eğitim verisi olarak kullanılması durumu.
4	A, B	A, B ve C'nin farklı kombinasyonlarda bir araya getirilmesiyle elde edilen birleştirilmiş veri setlerinin eğitim verisi olarak kullanılması durumu.
5	A, C	
6	B, C	
7	A, B, C	

Doc2vec yaklaşımı ile çeşitli veri setleri ve parametreler kullanılarak farklı modeller eğitilmiştir. Eğitim ve test kullanıcılarının temsil vektörlerinin elde edilmesinde sadece yazdıkları tweetler kullanılmıştır. Modelin eğitiminde kullanılan derlem çeşidine göre 7 farklı deney senaryosu belirlenmiştir. Bu deneyler Tablo 6.7'de verilmiştir. Tablo 6.7'de yer alan 7 senaryonun her biri Dm = 0/1 ve min count = 1/5 olmak üzere 4 farklı parametre seçimi üzerinden incelenerek toplam 28 deney gerçekleştirilmiştir. Bu parametrelerin açıklamaları Bölüm 5.4'te verilmiştir. Model eğitiminde vektör boyutu 100 olarak seçilmiştir. Daha önceki çalışmada bu parametrenin yöntemin başarısına olan etkisi detaylı olarak incelenmiştir. Bundan dolayı, bu çalışmada irdelenen parametrelerin daha rahat gözlemlenmesi için seçilen parametrelerin ayrıca vektör boyutu parametresine göre nasıl başarı gösterdiği araştırılmamıştır.

6.4.2. Doc2vec Yaklaşımının Metinsel Olmayan Bilgilerle Zenginleştirilmesi

Doc2vec, kelime temsil modeli olan word2vec yönteminin doküman temsillerinin elde edilmesi için geliştirilmiş versiyonudur. Doküman vektörlerinin elde edilmesinde giriş verisi metinsel verilerdir. Yani bir cümle, paragraf veya dokümanı oluşturan sözcük listesidir.

"Göztepe, Fenerbahçe'yi deplasmanda 2-1 mağlup etti" içeriğine sahip bir paylaşım; ekonomi, spor, politika kategorileri içerisinde spor kategorisine girer. "Böyle kargo firması olmaz, tercih ettiğim için pişmanım" içeriğine sahip bir paylaşım; pozitif, negatif, nötr duyguları içerisinde negatif olarak tanımlanır. Bu görevler için geliştirilen tahmin modellerinin dikkate alacağı başlıca özellikler (feature) kelimeler ve kelimelerin bir arada kullanım durumlarıdır. Paylaşımı gerçekleştiren kullanıcı profiline ait özelliklerin ve çeşitli yollarla (beğenme, yorum vb.) etkileşime geçen kullanıcı bilgilerinin modelin daha iyi performans göstermesine katkı sağlaması mümkün olsa da tez kapsamında ele alınan duygu sınıflandırma ve metin sınıflandırma problemlerinde bu bilgiler kullanılmamıştır. Her bir tweet bağımsız bir doküman olarak ele alınmıştır. Yüksek başarımlı derin öğrenme modelleri ortaya konulmuştur.

Bu tez çalışmasında sosyal medya kullanıcılarının temsil edilmesi denetimsiz bir öğrenme problemi olarak ele alınmıştır. Tweet temsillerinin bir araya getirilmesiyle kullanıcıların uzun bir zaman aralığında bahsettiği geniş konu yelpazesi kullanıcı vektörünün farklı boyutlarına gömülür. Bölüm 5'te verilen kullanıcı temsili yaklaşımında; Twitter kullanıcılarının gömmeleri, yazdıkları tweetlerin doküman vektörlerinin ortalamalarının alınması suretiyle elde edilmektedir. Bu bölümde yer alan doc2vec yaklaşımında (6.4.1) da bu yöntemin daha başarılı doküman temsilleri elde etmesi için farklı veri kaynaklarından elde edilen eğitim verileri ile kapsamlı bir deneysel inceleme gerçekleştirilmiştir.

Kullanıcı temsilleri elde edilirken tweet temsiliyle birlikte tweetin hangi kullanıcı tarafından yazıldığı, hangi kullancılar tarafından beğenildiği gibi çeşitli önemli bilgilerden faydalanılması amaçlanarak yeni bir yaklaşım önerilmiştir. Doküman modelinin metinsel

olmayan verilerle zenginleştirilmesi şeklinde tanımladığımız bu yaklaşımda kullanıcılarla ilgili çok daha fazla bilginin gömüldüğü yoğun vektörler elde edilmiştir.

Bu yaklaşımda tweet metni ile birlikte metinsel olmayan sosyal medya verilerinin de eğitim verisinde yer alması önerilmiştir. Tweet metni ile birlikte kullanıcıya (isim, konum, açıklama) ve tweetin etkileşimlerine (beğeni, paylaşım, yorum) ait farklı bilgiler doc2vec eğitim verisine eklenmiştir. Normalde sadece Tablo 6.8'de "tkn_1", "tkn_2", … olarak gösterilmiş olan tweet metninde yer alan kelimeler kullanılmaktadır. Bu yöntemde ise "loc_1", "desc_1", "Lusr_1" … gibi sözcükler eklenmiştir. Eklenen her bir bilgi için çevresine "nametag", "liketag" gibi kukla kelimeler (dummy token) eklenmiştir. Bununla ilgili detaylı açıklamalar Bölüm 6.3'te yer almaktadır.

Tablo 6.8. Farklı bilgilerle zenginleştirilmiş kullanıcı verisi örneği

Veri Türü	**Örnek**
Tweet Id	tweet_id_1
Kullanıcı adı	usr_name_1
Konum	loc_1
Açıklama	desc_1
Token listesi	tkn_1, tkn_2, tkn_3, tkn_4, tkn_5
Beğenenenler (like)	Lusr_1, Lusr_2, Lusr_3, Lusr_4, Lusr_5, Lusr_6
Paylaşanlar (RT)	Rusr_7, Rusr_8, Rusr_9, Rusr_10
Yorum yazanlar (comment)	Cusr_11, Cusr_12
Enriched_Data_1 (Zenginleştirilmiş Veri)	**tkn_1 tkn_2 tkn_3 tkn_4 tkn_5** enrtag enrtag enrtag enrtag enrtag nametag nametag nametag nametag nametag **usr_name_1** nametag nametag nametag nametag nametag loctag loctag loctag loctag loctag **loc_1** loctag loctag loctag loctag loctag desctag desctag desctag desctag desctag **desc_1** desctag desctag desctag desctag desctag liketag liketag liketag liketag liketag **Lusr_1** liketag liketag liketag liketag liketag **Lusr_2** liketag liketag liketag liketag liketag **Lusr_3** liketag liketag liketag liketag liketag **Lusr_4** liketag liketag liketag liketag liketag **Lusr_5** liketag liketag liketag liketag liketag **Lusr_6** liketag liketag liketag liketag liketag rttag rttag rttag rttag rttag rttag **Rusr_7** rttag rttag rttag rttag rttag **Rusr_8** rttag rttag rttag rttag rttag **Rusr_9** rttag rttag rttag rttag rttag **Rusr_10** rttag rttag rttag rttag rttag commtag commtag commtag commtag commtag **Cusr_11** commtag commtag commtag commtag **Cusr_12**
Tagged Document	*TaggedDocument ([Enriched_Data], [Tweet Id])*

Bu yöntemin uygulanması için D ve E veri setleri oluşturulmuştur. A veri setinde yer alan bir tweetin E veri setinde yer alan bilgilerle zengileştirilmesinin bir örneği Tablo 6.8'de verilmiştir. Tablo 6.9'da yer alan 5 farklı veri setinin kullanılması senaryosunun her biri Dm = 0/1 ve min count = 1/5 olmak üzere 4 farklı parametre seçimi üzerinden incelenerek toplam 20 deney gerçekleştirilmiştir. E veri seti aynı zamanda D'yi de kapsadığı için D veri seti ayrıca B ve C ile bir araya getirilerek bir deney gerçekleştirilmemiştir.

Tablo 6.9. Zenginleştirilmiş doküman modeli yaklaşımı (Deney senaryoları #2)

No	Veri seti	Senaryo açıklaması
1	A, D	Kullanıcı bilgilerinin tweet ile birleştirilmesi durumunun incelenmesi ve [97] çalışmasıyla karşılaştırılması.
2	A, E	Önerilen ana yaklaşım: tweet, kullanıcı bilgisi ve etkileşim bilgilerinin bir arada kullanılması.
3	A, E, B	Wikipedia ve genel amaçlı büyük tweet dokümanlarının önerilen ana yaklaşıma nasıl bir katkı sağlayacağının incelenmesi.
4	A, E, C	
5	A, E, B, C	

6.4.3. Kelime Gömme (PWE) Yöntemlerinin Uygulanması

Kelime gömmeleri birçok DDİ probleminde geleneksel kelime vektörlerine kıyasla daha iyi başarı göstererek şüphesiz son yılların en önemli gelişmelerinden birisi olmuştur. Önerdiğimiz PWE yaklaşımı; önceden eğitilmiş, hazır kelime gömmeleri $w \in \mathbb{R}^D$ kullanılarak her bir tweetin içerisinde geçen kelimelerin ortalaması şeklinde $\vec{T} \in \mathbb{R}^D$ temsil edilmesine dayanmaktadır. Önerilen yöntemler arasında en basit ve en az karmaşık olan bu yöntemle elde edilen tweet temsilleri kullanılarak kullanıcı temsilleri elde edilmiştir. Literatürde böyle bir kullanıcı temsil yöntemi bildiğimiz kadarıyla daha önce kullanılmamıştır. Metin temsil yöntemleri ile kullanıcı temsillerinin elde edilmesi alanında kelime veya doküman vektörü üzerinden önerilen bir yöntemin alt sınır olarak bu yöntemden daha iyi sonuç vermesinin beklenmesi doğru olacaktır.

Metin temsili konusunda yapılan çeşitli çalışmalarda asgari başarı beklentisi (baseline) ölçümünde yararlanmak üzere kelime gömmelerinin ortalamasının alınmasına ve TF-IDF gibi basit tekniklerin sonuçlarına yer verildiği görülmektedir [131,132,147]. Bu bölümde açıklanan PWE tabanlı teknikler üzerinden gerçekleştirilen deneyler de benzer bir amaçla tasarlanmıştır. Ayrıca en-gelişmiş (state-of-the-art) kelime gömme tekniklerinin kullanıcı temsillerinin elde

edilmesi konusunda detaylı bir karşılaştırması sunularak literatüre katkı sağlanmıştır. Tablo 6.10'da hazırlanan temel deneyler ve bu deneylerdeki ana amacın belirtildiği açıklamalar yer almaktadır. Bu bölümde gerçekleştirilen altıncı deney (TF-IDF) haricindeki diğer bütün deneylerde sadece A veri seti kullanılmıştır.

Tablo 6.10. PWE yaklaşımı (Deney senaryoları #3)

No	Yöntem	Senaryo açıklaması
1	Glove, FastText, Google News, Numberbatch	Hazır kullanılabilir (publicly available) PWE'lerin başarılarının karşılaştırması.
2	Glove	Farklı derlemler ile eğitilmiş Glove dil modellerinin başarılarının karşılaştırılması.
3	Word2Vec	Kendi oluşturduğumuz veri seti (A) ile eğitilen dil modelinin kullanımı. Farklı parametrelerin (vektör boyotu, minimum frekans değeri vb.) incelenmesi.
4	BERT, ELMO	Önceden eğitilmiş bağlamsal (contextual) metin temsil modellerinin kullanılması.
5	BERT, ELMO, Glove	Yığılmış gömme modellerinin (stacked embeddings) [37] kullanılması.
6	TF-IDF	İkincil bir başarı (baseline) testi için seyrek bir temsil yöntemi olan TF-IDF'in farklı terim (feature) sayıları ile araştırılması.

Önceden eğitilmiş kelime gömmelerinin kullanılması kapsamında Tablo 6.10'da yer alan deney senaryoları dışında farklı uygulamaların incelenmesi muhtemeldir. Aşağıda belirtilen senaryolar bu çalışmanın kapsamı dışında tutulmuştur:

- Kullanıma hazır dil modellerinden Glove, FastText, Google News gibi farklı seçenekler kullanılabilir. Bu yöntemlerle, kullanıcı tweetlerinden oluşan derlem üzerinden sıfırdan bir dil modeli eğitilerek kendi aralarında kapsamlı bir karşılaştırma yapılabilir.
- PWE'lerin kullanıcı tweetlerinden oluşan derlem üzerinden ince ayar (fine-tuning) yapıldıktan sonra kullanımı daha iyi sonuçlar verebilir.
- Bağlamsal metin temsillerinin kullanıcı temsili modellerinde kullanımıyla ilgili daha geniş bir araştırma yapılabilir. Bu çalışmada, yalnızca önceden eğitilmiş çok dilli versiyonları kullanılmıştır.

6.5. Sonuçlar

Üç ana başlık altında toplanan deney senaryolarının uygulanmasıyla elde edilen sonuçlar ortak performans metrikleri ile ayrı ayrı sunulmuştur. Kullanıcı gömmelerinin kalitesini belirlemek için göreve dayalı bir değerlendirme (task based evaluation) metodolojisi uygulanmıştır. Veri setlerinin ve model eğitim parametrelerinin etkisi ayrıntılı olarak verilmiştir.

Tablo 6.7, 6.9 ve 6.10'da verilen senaryolar için 5 gruba ait 500 Twitter kullanıcısı üzerinde deneyler gerçekleştirilmiştir. Bu kullanıcıların rasgele olarak %60'ı eğitim ($\boldsymbol{M = 300}$) ve %40'ı test ($\boldsymbol{N = 200}$) kullanıcıları olarak ayrılmıştır. Model başarılarının ölçülmesinde test kullanıcıları kullanılmıştır. Eğitim aşamasında test kullanıcılarına ait herhangi bir tweet, profil veya etkileşim bilgisi yer almamıştır.

Eğitim kullanıcılarının bilgileri iki şekilde kullanılmıştır. Bunlardan ilki, modelin eğitimi için eğitim veri setinde kullanımıdır. İkincisi ise, eğitim kullanıcılarının gömmeleri elde edilmek suretiyle ortalama grup gömmelerinin elde edilmesi şeklindedir. ***Deney senaryoları #1 ve #2***'de görüleceği üzere eğitim kullanıcıları hem model eğitiminde hem de ortalama grup gömmelerinin elde edilmesinde kullanılmıştır. ***Deney senaryoları #3*** kapsamında ise kullanıcı temsilleri doğrudan dil modelleri ile elde edildiğinden dolayı bir eğitim söz konusu değildir. Eğitim kullanıcıları sadece grup gömmelerinin elde edilmesi için kullanılmıştır. Bu deneylerden sadece TF-IDF yönteminde, kelime dağarcığının çıkarılması aşamasında eğitim kullanıcılarının tweetlerinden oluşan veri setine bağlı kalınmıştır. Bu şekilde tarafsız bir test uygulaması gerçekleştirilmiştir.

6.5.1. Deney Senaryoları #1 Sonuçları

Bu bölümde 6.4.1'de doc2vec yaklaşımı için açıklanan 28 deneysel durumun sonuçları Tablo 6.11, 6.12, 6.13 ve 6.14'te sunulmuştur.

Kullanılan eğitim veri setine göre her bir modelin kullanıcı temsili elde etme görevindeki başarısı ayrı ayrı verilmiştir. Genel başarı olarak doğruluk (accuracy) değeri kullanılmıştır. Kesinlik (precision), duyarlılık (recall) ve F1 skorları her bir grup için ayrı ayrı gösterilmiştir.

Deney senaryoları #1 ve Deney senaryoları #2'de sunulan modeller 20 iterasyon boyunca eğitilmiştir. Her bir iterasyon sonunda modeller kaydedilerek başarıları ölçülmüştür. Verilen sonuçlar her bir deneyin 20 iterasyon içerisinde en yüksek başarıyı gösteren modele aittir. Modeller ayrıca aynı parametreler ile 100 iterasyon boyunca her 10 iterasyonda bir ve 2000 iterasyon boyunca her 100 iterasyonda bir kaydedilmek suretiyle test edilmiştir. Bu bölümde sunulan sonuçların paralelinde performans değerleri elde edildiğinden dolayı bu sonuçlara ayrıca yer verilmemiştir.

Tablo 6.11. Deney Senaryoları #1 Sonuçları (Dm=0, MinCount=1)

Eğitim Verisi	Grup	Kesinlik	Duyarlılık	F1	Doğruluk
A	G1	0.971	0.825	0.892	0.910
	G2	0.974	0.925	0.949	
	G3	0.971	0.825	0.892	
	G4	0.741	1.000	0.851	
	G5	0.975	0.975	0.975	
B	G1	0.919	0.850	0.883	**0.935**
	G2	0.946	0.875	0.909	
	G3	0.907	0.975	0.940	
	G4	0.952	1.000	0.976	
	G5	0.951	0.975	0.963	
C	G1	0.971	0.825	0.892	0.910
	G2	0.971	0.850	0.907	
	G3	0.809	0.950	0.874	
	G4	0.950	0.950	0.950	
	G5	0.886	0.975	0.929	
A + B	G1	0.917	0.825	0.868	0.915
	G2	0.872	0.850	0.861	
	G3	0.884	0.950	0.916	
	G4	0.974	0.950	0.962	
	G5	0.930	1.000	0.964	
B + C	G1	0.889	0.800	0.842	0.900
	G2	0.846	0.825	0.835	
	G3	0.881	0.925	0.902	
	G4	0.952	1.000	0.976	
	G5	0.927	0.950	0.938	
A + C	G1	0.970	0.800	0.877	0.925
	G2	0.974	0.950	0.962	
	G3	0.816	1.000	0.899	
	G4	0.974	0.925	0.949	
	G5	0.927	0.950	0.938	
A + B + C	G1	0.914	0.800	0.853	0.915
	G2	0.872	0.850	0.861	
	G3	0.870	1.000	0.930	
	G4	0.974	0.950	0.962	
	G5	0.951	0.975	0.963	

Tablo 6.12. Deney Senaryoları #1 Sonuçları (Dm=0, MinCount=5)

Eğitim Verisi	Grup	Kesinlik	Duyarlılık	F1	Doğruluk
A	G1	0.968	0.750	0.845	0.925
	G2	0.929	0.975	0.951	
	G3	1.000	0.900	0.947	
	G4	0.800	1.000	0.889	
	G5	0.976	1.000	0.988	
B	G1	0.923	0.900	0.911	0.945
	G2	0.972	0.875	0.921	
	G3	0.929	0.975	0.951	
	G4	0.952	1.000	0.976	
	G5	0.951	0.975	0.963	
C	G1	0.970	0.800	0.877	0.915
	G2	0.923	0.900	0.911	
	G3	0.844	0.950	0.894	
	G4	0.974	0.950	0.962	
	G5	0.886	0.975	0.929	
A + B	G1	0.969	0.775	0.861	0.935
	G2	0.826	0.950	0.884	
	G3	0.950	0.950	0.950	
	G4	0.952	1.000	0.976	
	G5	1.000	1.000	1.000	
B + C	G1	0.971	0.825	0.892	0.940
	G2	0.923	0.900	0.911	
	G3	0.907	0.975	0.940	
	G4	0.952	1.000	0.976	
	G5	0.952	1.000	0.976	
A + C	G1	0.971	0.850	0.907	0.930
	G2	0.927	0.950	0.938	
	G3	0.864	0.950	0.905	
	G4	0.974	0.925	0.949	
	G5	0.929	0.975	0.951	
A + B + C	G1	0.973	0.900	0.935	**0.965**
	G2	0.974	0.950	0.962	
	G3	0.951	0.975	0.963	
	G4	0.976	1.000	0.988	
	G5	0.952	1.000	0.976	

Tablo 6.13. Deney Senaryoları #1 Sonuçları (Dm=1, MinCount=1)

Eğitim Verisi	Grup	Kesinlik	Duyarlılık	F1	Doğruluk
A	G1	0.923	0.900	0.911	0.945
	G2	1.000	0.975	0.987	
	G3	0.946	0.875	0.909	
	G4	0.889	1.000	0.941	
	G5	0.975	0.975	0.975	
B	G1	0.892	0.825	0.857	0.920
	G2	0.865	0.800	0.831	
	G3	0.886	0.975	0.929	
	G4	0.976	1.000	0.988	
	G5	0.976	1.000	0.988	
C	G1	1.000	0.800	0.889	0.940
	G2	0.950	0.950	0.950	
	G3	0.905	0.950	0.927	
	G4	0.952	1.000	0.976	
	G5	0.909	1.000	0.952	
A + B	G1	0.972	0.875	0.921	0.930
	G2	1.000	0.875	0.933	
	G3	0.780	0.975	0.867	
	G4	1.000	0.925	0.961	
	G5	0.952	1.000	0.976	
B + C	G1	0.906	0.725	0.806	0.890
	G2	0.821	0.800	0.810	
	G3	0.822	0.925	0.871	
	G4	0.930	1.000	0.964	
	G5	0.976	1.000	0.988	
A + C	G1	0.974	0.950	0.962	**0.970**
	G2	1.000	0.925	0.961	
	G3	0.929	0.975	0.951	
	G4	0.976	1.000	0.988	
	G5	0.976	1.000	0.988	
A + B + C	G1	0.972	0.875	0.921	0.940
	G2	1.000	0.925	0.961	
	G3	0.830	0.975	0.897	
	G4	1.000	0.925	0.961	
	G5	0.930	1.000	0.964	

Tablo 6.14. Deney Senaryoları #1 Sonuçları (Dm=1, MinCount=5)

Eğitim Verisi	Grup	Kesinlik	Duyarlılık	F1	Doğruluk
A	G1	0.973	0.900	0.935	0.960
	G2	1.000	0.975	0.987	
	G3	0.949	0.925	0.937	
	G4	0.909	1.000	0.952	
	G5	0.976	1.000	0.988	
B	G1	0.938	0.750	0.833	0.900
	G2	0.773	0.850	0.810	
	G3	0.864	0.950	0.905	
	G4	0.974	0.950	0.962	
	G5	0.976	1.000	0.988	
C	G1	1.000	0.850	0.919	0.945
	G2	0.949	0.925	0.937	
	G3	0.884	0.950	0.916	
	G4	0.952	1.000	0.976	
	G5	0.952	1.000	0.976	
A + B	G1	0.971	0.850	0.907	0.935
	G2	0.974	0.925	0.949	
	G3	0.800	1.000	0.889	
	G4	1.000	0.925	0.961	
	G5	0.975	0.975	0.975	
B + C	G1	0.972	0.875	0.921	0.955
	G2	0.949	0.925	0.937	
	G3	0.929	0.975	0.951	
	G4	1.000	1.000	1.000	
	G5	0.930	1.000	0.964	
A + C	G1	0.974	0.950	0.962	**0.975**
	G2	1.000	0.950	0.974	
	G3	0.951	0.975	0.963	
	G4	1.000	1.000	1.000	
	G5	0.952	1.000	0.976	
A + B + C	G1	0.974	0.950	0.962	**0.975**
	G2	1.000	0.950	0.974	
	G3	0.951	0.975	0.963	
	G4	1.000	1.000	1.000	
	G5	0.952	1.000	0.976	

Dağıtık doküman temsili (doc2vec) yaklaşımının çok sayıda tweet içeren genel amaçlı bir veri seti (*C* veri seti) ile eğitilen model ile twitter kullanıcılarının temsili görevinde %94.5 başarı sağlayabildiği görülmüştür. *A* veri setinde olduğu gibi söz konusu probleme özgü, özel seçilen kullanıcılara ait tweetlerden oluşan bir derlem tercih edildiğinde başarı %96'ya yükselebilmektedir. En yüksek başarı ise *A* ve *C* veri setlerinin bir araya getirilmesi ile %97.5 olarak elde edilmiştir. *A, B* ve *C* veri setlerinin bir araya getirildiği deneyde yine %97.5 başarı elde edilmiştir. Fakat, yalnızca *B* ve *A+B* veri setleri ile elde edilen sonuçlara dikkat edildiğinde *B* veri setinin bu yüksek başarıda etkili olmadığı anlaşılmaktadır. Dolayısıyla Twitter alanı (domain) dışından, Wikipedia gibi kaynakların derlem olarak kullanılmasının kullanıcı temsillerinin elde edilmesinde iyi bir tercih olmadığı tespit edilmiştir.

%97.5 doğruluk oranı ile en iyi sonucu veren modelin hata matrisi (confusion matrix) Tablo 6.15'te verilmiştir.

Tablo 6.15. Deney senaryoları #1 hata matrisi

	G1	G2	G3	G4	G5
G1	38	0	1	0	1
G2	1	38	1	0	0
G3	0	0	39	0	1
G4	0	0	0	40	0
G5	0	0	0	0	40

6.5.2. Deney Senaryoları #2 Sonuçları

Bu bölümde 6.4.2'de doc2vec yaklaşımının metinsel olmayan bilgilerle zenginleştirilmesi yaklaşımı için açıklanan 20 deneysel durumun sonuçları Tablo 6.16, 6.17, 6.18, 6.19'da sunulmuştur.

Tablo 6.16. Deney Senaryoları #2 Sonuçları (Dm=0, MinCount=1)

Eğitim Verisi	Grup	Kesinlik	Duyarlılık	F1	Doğruluk
A + D	G1	1.000	0.625	0.769	0.880
	G2	1.000	0.900	0.947	
	G3	0.841	0.925	0.881	
	G4	0.780	0.975	0.867	
	G5	0.867	0.975	0.918	

A + E	G1	0.897	0.875	0.886	**0.930**
	G2	1.000	0.950	0.974	
	G3	0.826	0.950	0.884	
	G4	0.974	0.950	0.962	
	G5	0.974	0.925	0.949	
A + B + E	G1	0.925	0.925	0.925	**0.930**
	G2	0.974	0.950	0.962	
	G3	0.900	0.900	0.900	
	G4	0.923	0.900	0.911	
	G5	0.929	0.975	0.951	
A + C + E	G1	0.895	0.850	0.872	0.905
	G2	0.975	0.975	0.975	
	G3	0.800	0.900	0.847	
	G4	0.881	0.925	0.902	
	G5	1.000	0.875	0.933	
A + B + C + E	G1	0.837	0.900	0.867	0.890
	G2	0.974	0.925	0.949	
	G3	0.875	0.875	0.875	
	G4	0.895	0.850	0.872	
	G5	0.878	0.900	0.889	

Tablo 6.17. Deney Senaryoları #2 Sonuçları (Dm=0, MinCount=5)

Eğitim Verisi	**Grup**	**Kesinlik**	**Duyarlılık**	**F1**	**Doğruluk**
A + D	G1	0.900	0.675	0.771	0.895
	G2	0.884	0.950	0.916	
	G3	0.854	0.875	0.864	
	G4	0.870	1.000	0.930	
	G5	0.975	0.975	0.975	
A + E	G1	0.750	0.750	0.750	0.860
	G2	1.000	0.950	0.974	
	G3	0.776	0.950	0.854	
	G4	0.860	0.925	0.892	
	G5	0.967	0.725	0.829	

A + B + E	G1	0.907	0.975	0.940	**0.960**
	G2	1.000	0.975	0.987	
	G3	1.000	0.925	0.961	
	G4	0.951	0.975	0.963	
	G5	0.950	0.950	0.950	
A + C + E	G1	0.884	0.950	0.916	0.945
	G2	1.000	0.975	0.987	
	G3	0.927	0.950	0.938	
	G4	0.950	0.950	0.950	
	G5	0.973	0.900	0.935	
A + B + C + E	G1	0.902	0.925	0.914	0.940
	G2	1.000	0.975	0.987	
	G3	0.950	0.950	0.950	
	G4	0.902	0.925	0.914	
	G5	0.949	0.925	0.937	

Tablo 6.18. Deney Senaryoları #2 Sonuçları (Dm=1, MinCount=1)

Eğitim Verisi	**Grup**	**Kesinlik**	**Duyarlılık**	**F1**	**Doğruluk**
A + D	G1	0.902	0.925	0.914	0.925
	G2	0.974	0.950	0.962	
	G3	0.939	0.775	0.849	
	G4	0.909	1.000	0.952	
	G5	0.907	0.975	0.940	
A + E	G1	0.884	0.950	0.916	**0.940**
	G2	1.000	0.950	0.974	
	G3	0.974	0.925	0.949	
	G4	0.889	1.000	0.941	
	G5	0.972	0.875	0.921	
A + B	G1	0.917	0.825	0.868	0.920
	G2	1.000	0.950	0.974	
	G3	0.947	0.900	0.923	

+	G4	0.800	1.000	0.889	
E	G5	0.974	0.925	0.949	
A	G1	0.927	0.950	0.938	
+	G2	1.000	0.950	0.974	
C	G3	0.974	0.950	0.962	**0.940**
+	G4	0.851	1.000	0.920	
E	G5	0.971	0.850	0.907	
	G1	0.881	0.925	0.902	
A + B	G2	1.000	0.950	0.974	
+	G3	0.974	0.925	0.949	0.935
C + E	G4	0.870	1.000	0.930	
	G5	0.972	0.875	0.921	

Tablo 6.19. Deney Senaryoları #2 Sonuçları (Dm=1, MinCount=5)

Eğitim Verisi	**Grup**	**Kesinlik**	**Duyarlılık**	**F1**	**Doğruluk**
	G1	0.895	0.850	0.872	
A	G2	0.974	0.925	0.949	
+	G3	0.939	0.775	0.849	0.910
D	G4	0.833	1.000	0.909	
	G5	0.930	1.000	0.964	
	G1	0.950	0.950	0.950	
A	G2	1.000	0.975	0.987	
+	G3	0.975	0.975	0.975	**0.970**
E	G4	0.952	1.000	0.976	
	G5	0.974	0.950	0.962	
A	G1	0.946	0.875	0.909	
+	G2	1.000	0.950	0.974	
B	G3	0.974	0.925	0.949	0.950
+	G4	0.870	1.000	0.930	
E	G5	0.976	1.000	0.988	
A	G1	0.951	0.975	0.963	
+	G2	1.000	0.950	0.974	
C	G3	0.973	0.900	0.935	0.960
+	G4	0.909	1.000	0.952	
E	G5	0.975	0.975	0.975	

A + B + C + E	G1	0.921	0.875	0.897	0.950
	G2	1.000	0.950	0.974	
	G3	0.974	0.950	0.962	
	G4	0.889	1.000	0.941	
	G5	0.975	0.975	0.975	

Çalışmanın ana veri seti olan *A*'nın *E* veri seti ile zenginleştirilmesi sonucunda maksimum başarı %96'dan %97'ye yükselmiştir. Fakat *A*'nın *D* veri seti ile zenginleştirilmesinin başarıyı yükseltmediği tespit edilmiştir. Yani, kullanıcı profil bilgileri tek başına yeterli olmayıp ancak aktivite bilgileriyle bir arada kullanıldığında performansı arttırmaktadır.

%97 doğruluk oranı ile en iyi sonucu veren modelin hata matrisi (confusion matrix) Tablo 6.20'de verilmiştir.

Tablo 6.20. Deney senaryoları #2 hata matrisi

	G1	G2	G3	G4	G5
G1	38	0	1	0	1
G2	1	39	0	0	0
G3	0	0	39	1	0
G4	0	0	0	40	0
G5	1	0	0	1	38

Doc2vec yaklaşımının önerildiği modellerin başarısı, kullanıcıların eğitim ve test kümelerinde farklı oranlarda (%80-%20, %70-%30, %50-%50) yer aldığı durumlar için de incelenmiştir. Verilen doğruluk değerlerinden farklı sonuçlar elde edilse de modeller arasındaki performans sıralamasında dikkate değer bir değişiklik gözlenmemiştir. Eğitim ve test kümeleri dengesiz (imbalanced) bir yapıda kurgulandığında yine aynı durum söz konusu olmuştur. Bu farklı dağılımlara ait sonuçlara çalışmanın kapsamını amacın dışında genişleteceğinden dolayı ayrıca verilmemiştir. Zira, sadece Deney senaryoları #1 ve #2 için mevcut veri seti dağılımları ile toplam 48 farklı deney koşturulmuştur.

6.5.3. Deney Senaryoları #3 Sonuçları

Bu bölümde 6.4.3'de açıklanan deneylerin sonuçları sunulmuştur. Deney senaryoları #3'ün 4 ve 5 numaralı deneyleri bir arada olmak üzere diğer her bir deney için sonuçlar ayrı birer tablo halinde verilmiştir. Asgari başarı (baseline) değerlendirmesi amacıyla uygulanan bu senaryo deneylerinde doğruluk ve F1 skorlarının verilmesi yeterli görülmüştür. Kesinlik ve duyarlılık

değerleri verilmeyerek daha az karmaşık bir karşılaştırma ortamı sunulmuştur. Gruplar için ayrı ayrı elde edilen F1 skorlarının ağırlıklı ortalaması kullanılmıştır.

Tablo 6.21'de 1 numaralı deneyde belirtilen farklı PWE modellerinin başarıları yer almaktadır. Glove için elde edilen sonuçlar Tablo 6.22'de daha detaylı olarak bulunmaktadır. Google News gömmelerinin diğer hazır PWE'ler kadar iyi sonuç vermediği görülmüştür.

Tablo 6.21. PWE modellerinin karşılaştırılması

Model	Gömme Boyutu	Doğruluk	F1
FastText	300	0.810	**0.809**
Numberbatch	300	0.810	0.807
Google News	300	0.720	0.713

Glove modelinin hazır kelime gömmeleri farklı vektör boyutlarındadır [148]. Ayrıca eğitildiği derlem bakımından farklı çeşitleri bulunmaktadır. 2 numaralı deneyde eğitim verisi kaynağına göre Twitter ve Wikipedia metinlerinden elde edilmiş derlemlerin karşılaştırılması yapılmıştır. Sözlük boyutu, token sayısı ve gömme boyutu parametrelerine göre detaylı bir inceleme yapılmıştır. En iyi sonucu tweetlerden oluşan bir derlemde eğitilmiş 200 boyutlu hazır Glove gömmeleri sağlamıştır.

Tablo 6.22. Glove hiper parametre incelemesi

Derlemin Kaynağı	Sözlük Boyutu	Token Sayısı	Gömme Boyutu	Doğruluk	F1
Wikipedia	400×10^3	6×10^9	100	0.835	0.828
			200	0.815	0.808
			300	0.800	0.797
Twitter	1200×10^3	27×10^9	25	0.800	0.799
			50	0.805	0.799
			100	0.835	0.830
			200	**0.840**	**0.834**

FastText, Numberbatch, Google News ve Glove hazır kelime gömmelerinin incelenmesi sonrasında *A* veri seti üzerinde, sıfırdan eğitilmiş bir word2vec modelinin nasıl başarı göstereceği araştırılmıştır (3 numaralı deney). 100 boyutlu sabit bir vektör uzunluğu için farklı "minimum count" parametreleri için eğitilen kelime vektörleri kullanılarak elde edilen başarılar Tablo 6.23'te verilmiştir.

Tablo 6.23. Word2vec kullanıcı temsili deneyleri

Min Count	Doğruluk	F1
1	0.790	0.786
3	0.830	0.827
5	0.845	**0.842**

Kendi eğittiğimiz word2vec kelime gömmelerinin bütün hazır PWE'lerden daha iyi sonuç verdiği görülmüştür.

Dördünce deneyde; önceden eğitilmiş bağlamsal (contextual) metin temsillerinin kelime temsil modellerinin gelenekselleşen word2vec türevi kelime gömmelerine göre bir üstünlük sağlayıp sağlamayacağı araştırılmıştır. Beşinci deneyde ise hazır kelime gömmeleri arasında en iyi performansı gösteren Glove modeli ile Bert ve Elmo modellerinin bir araya getirilmesiyle elde edilen 6244 boyutlu bir vektörle kullanıcı temsilleri elde edilerek başarısı test edilmiştir. Bu iki deneye ait sonuçlar Tablo 6.24'te verilmiştir. Yığılmış gömme modellerinin (stacked embeddings) kullanılmasında Flair [52] tarafından sunulan kütüphaneden yararlanılmıştır. Flair ile yığılmış doküman vektörü elde etmek için temel kod adımları Şekil 6.6'da verilmiştir.

```
from flair.embeddings import Sentence
from flair.embeddings import WordEmbeddings
from flair.embeddings import ELMoEmbeddings
from flair.embeddings import BertEmbeddings
from flair.embeddings import DocumentPoolEmbeddings

# gömme modellerinin yüklenmesi
glove_embedding = WordEmbeddings('glove')
bert_embedding = BertEmbeddings()
elmo_embedding = ELMoEmbeddings()

# gömme modelini birleştirerek doküman gömmeleri oluştur
document_embeddings = DocumentPoolEmbeddings([glove_embedding,
                                              bert_embedding,
                                              elmo_embedding])


# ön işleme aşamaları tamamlanmış bir dokümanın temsili
dokuman = Sentence(txt)
document_embeddings.embed(dokuman)
```

Şekil 6.6. Flair ile yığılmış doküman vektörü elde edilmesi

Bert modelinin FastText, Numberbatch, Google News'ten daha iyi sonuç verdiği görülmüştür. Elmo ise aynı başarıyı göstermemiştir. Yığılmış doküman vektörlerinin tekil olarak Bert, Elmo ve Glove'un kullanılmasına göre bir katkısı olmamıştır.

Bert modelinin hiçbir iyileştirme (fine-tuning) aşamasından geçirilmeden, çok dilli (multilingual) bir versiyonunun diğer hazır temsillerden daha iyi sonuç vermesi, kullanıcı temsili görevinde gelecek vaat eden bir yaklaşım olduğunu göstermektedir.

Tablo 6.24. Bağlamsal metin temsillerinin kullanılması

Model	Gömme Boyutu	Doğruluk	F1
Bert	3072	0.815	**0.814**
Elmo	3072	0.770	0.765
Bert-Elmo-Glove (Stacked)	3072 + 3072 + 100	0.815	0.812

Altıncı deneyde TF-IDF ile kullanıcı temsillerinin elde edilmesi gerçekleştirilmiştir. Elde edilen sonuçlar Tablo 6.25'te bulunmaktadır. Kullanıcı temsili görevinde TF-IDF yönteminin kullanılmasındaki en kritik faktörün vektör boyutu olduğu görülmüştür. 100 ile 100.000 arasında farklı boyut değerleri seçilmiştir. 5 farklı veri seti için 6 farklı vektör boyutu değerinin ayrı ayrı denenmesi sonucunda 30 farklı deney gerçekleştirilmiştir.

Tablo 6.25. TF-IDF ile kullanıcı temsili sonuçları

Maksimum özellik sayısı	Veri setine bazında başarısı									
	A		B		C		A, D		A, E	
	Doğ.	F1	Doğ.	F1	Doğ.	F1	Doğ.	F1	Doğ.	F1
100	0.615	0.616	0.580	0.588	0.495	0.482	0.595	0.572	0.490	0.478
1000	0.865	0.866	0.830	0.861	0.760	0.764	0.615	0.601	0.500	0.487
10000	0.905	0.906	0.540	0.534	0.875	0.873	0.695	0.685	0.500	0.489
20000	0.915	0.916	0.575	0.574	0.875	0.873	0.660	0.648	0.490	0.481
50000	0.915	0.916	0.590	0.590	0.900	0.900	0.645	0.639	0.490	0.481
100000	0.915	0.916	0.675	0.674	0.900	0.900	0.665	0.659	0.490	0.481

TF-IDF yöntemindeki maksimum özellik sayısı değeri doğrudan vektör boyutu büyüklüğünü belirlemektedir. Söz konusu derlem için (A, B, C, … veri setleri) öncelikle terim frekanslarının hesaplanması gerekir. Bu aşamada en fazla kaç farklı terimin ağırlıklandırılacağına

karar verilmelidir. Bu değerin, diğer deneylerdeki kullanıcı gömme boyutlarına paralel olarak 100, 200, 300 gibi sayılarda belirlendiğinde %61,5 ve daha düşük başarı sağlamıştır. Daha yüksek boyutlarda performansın nasıl etkilendiği detaylı olarak ele alınmıştır. Ancak 10.000 ve daha büyük sayıda farklı terim kullanıldığında verebileceği en başarılı sonuçlara ulaşmaktadır.

Veri seti bakımından bu yöntemin başarısı değerlendirildiğinde sadece *A* ve *C* veri setlerinin iyi sonuç verdiği görülmüştür. *D* ve *E* veri setleriyle uygulanan sosyal medya verilerinin zenginleştirilmesi yaklaşımında, TF-IDF yönteminin bu özelliklerin katkısını değerlendiremediği sonucu ortaya çıkmıştır. Bu veri setlerinin, herhangi bir maksimum-özellik-sayısı için *A* ve *C* veri setlerinin kullanılmasıyla elde edilen başarıya yaklaşamadığı Tablo 6.25'te görülmektedir.

A ve C veri setlerinin performansları kendi aralarında karşılaştırıldığında ise A'nın eğitim verisi olarak kullanılmasının daha iyi sonuç verdiği görülmüştür. Yani, bu görev için özel olarak seçilmiş 300 kullanıcıya ait 300.000 adet tweet, genel amaçlı bir tweet veri setinde yer alan bir milyon tweete göre daha elverişlidir.

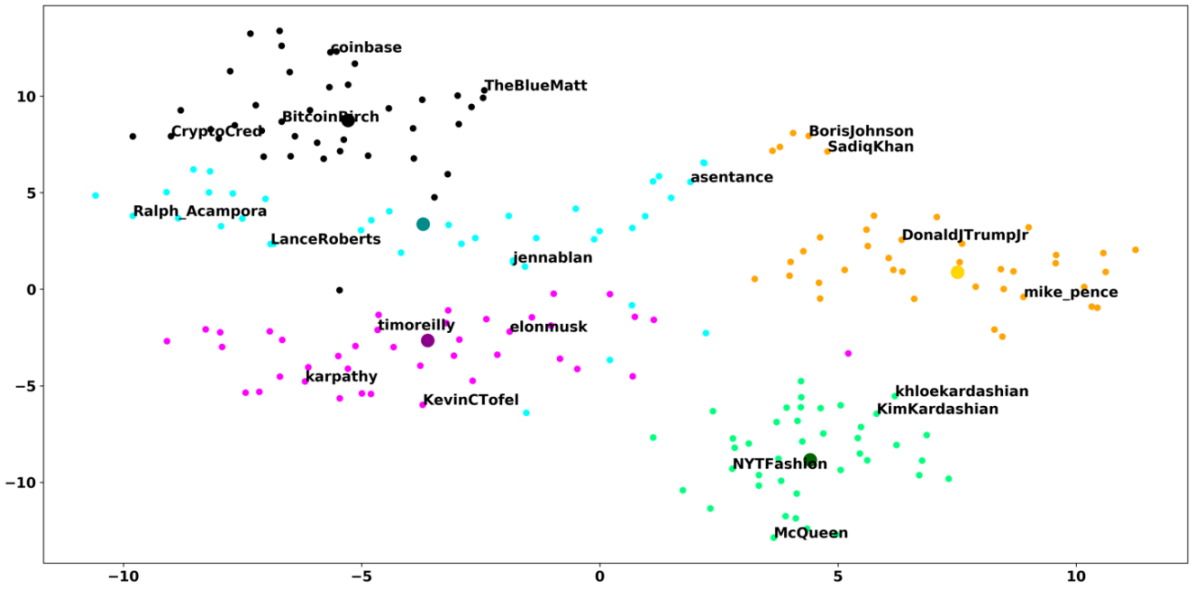
Genel olarak hazır kelime gömmelerinin ve TF-IDF yaklaşımının önerilen doc2vec yöntemi ve bu yöntemin farklı sosyal medya verileriyle zenginleştirilmiş veri setlerinden yararlanmasına yönelik geliştirdiğimiz versiyonundan daha iyi sonuç vermediği ortaya çıkmıştır.

6.6. Pratik Uygulamalar

Bu bölümde, kullanıcı temsil modelinin kullanım senaryolarına yer verilmiştir. Hem bilimsel çalışmalarda hem de ticari dünyada çok geniş bir yelpazede yoğun kullanıcı vektörlerinden yararlanılabilir. Önerilen yöntemin, kullanıcılar arasında benzerlik hesaplamalarına olanak sağlaması ve onları vektör uzayında konumlandırabilmesi üzerinden çeşitli örnekler sunulmuştur.

Elde edilen kullanıcı temsillerinin Şekil 6.7'de verilen örnekteki gibi görselleştirilmesi ile çeşitli analizler yapılabilir. Veri setinin test kümesinde bulunan kullanıcıların 100 boyutlu gömmeleri t-SEN [149] ile iki boyutlu uzayda birer noktaya dönüştürülmüştür. Bu noktalar kullanıcıların bilinen grup aidiyetlerine göre renklendirilmiştir. Beş grubun ortalama temsilleri daha koyu renkli ve daha büyük noktalarla işaretlenmiştir. Her gruptan dörder kullanıcının bulundukları konumların üzerine kullanıcı isimleri yazdırılmıştır. Kullanılan veri setinin 2018 yılı ve daha öncesini kapsadığı bilgisi göz önünde bulundurularak, kullanıcıların gerçek Twitter profilleri incelendiğinde, birbirine yakın konumlandırılmış kullanıcılar arasında Twitter dünyasında ve hatta gerçek hayatta fark edilir benzerlikler olduğu görülmektedir. Bu benzerlikler için verilebilecek bazı örnekler şu şekildedir:

- Şekil 6.7 incelendiğinde turuncu ile renklendirilmiş "Amerikan veya İngiliz Siyaseti" temalı gruba ait kullanıcıların diğer gruplardan yüksek oranda ayrıştığı görülecektir.
- Cumhuriyetçi Parti (Amerika Birleşik Devletleri) ile alakalı "*donaldjtrumpjr*" ve "*mike_pence*" kullanıcı adlı kişiler birbirine yakın olup "*BorisJohnson*" ve "*SadiqKhan*" kullanıcı isimlerine sahip İngiltere siyasetinden kişilere uzaktır. Siyaset ortak noktasında bir arada bulunan bu 4 Twitter kullanıcısının 2D bir çizimde bu yakınlıklarda konumlanması yöntemin başarısını göstermektedir.
- "Yaşam Tarzı ve Moda" temalı gruba ait kullanıcılardan "*khloekardashian*" ve "*KimKardashian*" kullanıcı isimli kişiler birbirine çok yakın konumlanmıştır. Gerçek hayatta gerçekten de moda ve yaşam tarzı konularıyla alakalı olduğu bilinen bu kişiler aynı zamanda kardeştir.

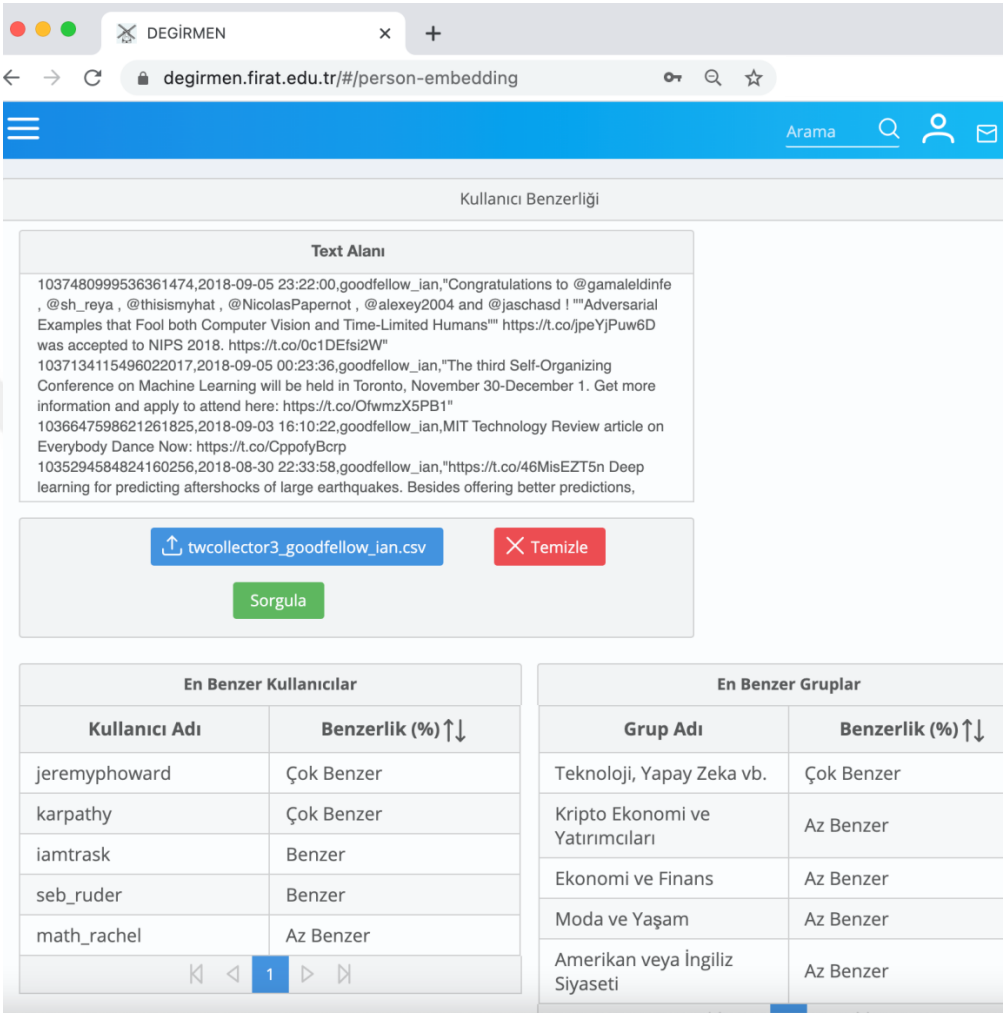


Şekil 6.7. En başarılı modele dayalı 2 boyutlu kullanıcı temsilleri

Kullanıcı temsilleriyle gerçekleştirilebilecek bir diğer analiz ise doğrudan benzer kullanıcıların tespit edilmesidir. Savunma Sanayii Müsteşarlığı tarafından desteklenen, Fırat Üniversitesi Büyük Veri ve Yapay Zekâ Laboratuvarı'nda yürütülen "Derin Öğrenme Büyük Veri Analizi Platformu (DEĞİRMEN)" projesinin sosyal medya analizi ve eğilim tespiti çalışmaları kapsamında önerdiğimiz kullanıcı temsil modelinden yararlanılmıştır.

Şekil 6.8'de DEĞİRMEN projesinin portal sayfasından kullanıcı eğilim analizi için sunulan bir demo ara yüzü bulunmaktadır. Bu ara yüzde verileri dışardan aktarılan bir kullanıcıya en benzer kullanıcılar listelenmektedir. Ayrıca önceden tanımlı gruplardan hangileriyle ne kadar

alakalı olduğu ortalama grup temsilleriyle olan benzerlik üzerinden sunulmaktadır. Bu şekilde, daha önceden temsil edilmiş diğer kullanıcılar üzerinden muhakeme imkânı sunan, eğilim analizi için yardımcı bir araç geliştirilmiştir.



Şekil 6.8. Kullanıcı benzerliği portal ara yüzü

Bu sistemde, sorgulanan kullanıcının verileriyle 100 boyutlu bir gömme elde edildikten sonra veri tabanında bulunan diğer bütün kullanıcıların gömmeleriyle Öklid uzaklığı hesaplanır. Uzaklık değerleri küçükten büyüğe doğru sıralanarak en yakın 5 tanesi gösterilir. Web portal ara yüzünde gösterildiği gibi "çok benzer", "benzer", "az benzer" şeklinde bütün uzaklıklar üzerinden bir ölçekleme yapılabilir.

Şekil 6.9'da web portal arka planında çalışan uygulama kodları tanıtılmıştır. Bir kullanıcının ("$JoeBiden$") gömme değerinin elde edilmesi ("$user2vec$") ve en benzer kullanıcıların listelenmesi ("$similar\_users$") için hazırlanan metotlar görülmektedir. Şekil 6.10'da grup benzerlikleri için aynı işlevler gösterilmiştir.

```
# JoeBiden isimli kullanıcının verilerini oku
kullanici_verisi = os.path.join(test_users_path,"twcollector5_JoeBiden.csv")

# bu CSV dosyasında tweetleri olan kullanıcının embedding değerini çıkar
kullanici_vektoru = us.user2vec(kullanici_verisi)

# hazırladığımız "similar_users" metoduyla en yakın N tane kullanıcı ve uzaklıkları
kullanicilar = us.similar_users(kullanici_vektoru, 5)

#göster
for kullanici in kullanicilar:
    print(kullanici, ": {:.2f}".format(kullanicilar[kullanici]))
```

```
MittRomney : 3.91
alfranken : 4.54
HillaryClinton : 4.97
BarackObama : 5.21
CoryBooker : 5.63
```

Şekil 6.9. User2vec kullanıcı benzerliği kodları

```
In [49]: # kullanıcının alakalı olduğu (en çok benzediği) grubu bul
         group = us.most_similar_group(kullanici_vektoru)
         print(group)

         Amerikan veya İngiliz Siyaseti

In [50]: # grupların kullanıcıya olan yakınlıklarını bul
         groups = us.similar_groups(kullanici_vektoru, 5)

         for group in groups:
             print(group, ": {:.2f}".format(groups[group]))

         twcollector5 : 4.71
         twcollector3 : 7.11
         twcollector4 : 7.30
         twcollector1 : 7.31
         twcollector2 : 7.88
```

Şekil 6.10. User2vec grup benzerliği kodları

6.7. Fiziksel Ortam ve Hesaplama Performansı

Bu bölümde deneylerin gerçekleştirildiği fiziksel hesaplama ortamı ve kaydedilen çalışma süreleri gibi detaylara yer verilmiştir. Gerçekleştirilen onlarca deneyin hemen hemen hepsi hesaplama yoğun (compute-intensive) görevlerdir. Modellerin en başarılı versiyonlarının bulunması ve birbirleriyle karşılaştırılması için büyük miktarda bellek ve hesaplama kapasitesi kullanılmıştır. Spesifik olarak Gensim kütüphanesine bağlı olarak açıklanan süreçlerle ilgili açıklamalar Bölüm 5.4.'te yer almaktadır.

Doküman modellerinin başarısı, eğitim süresince her bir iterasyon sonunda ölçülmüştür. Test aşamasında 500 kullanıcıya ait yalın veya farklı bilgilerle zenginleştirilmiş tweet temsilleri elde edilmektedir. Modellerin eğitiminden daha fazla kaynak test süreçleri için kullanılmıştır.

Doküman tabanlı kullanıcı temsil modelinin test edilmesi, her bir kullanıcı için 1000 adet kaydın temsil vektörlerinin çıkarılmasını gerektirir. Kayıtlar metinsel veriler olup; tweet, kullanıcı

bilgileri ve etkileşim verilerinden bir veya birkaçının bir araya getirilmesinden meydana gelmektedir. Eğitilmiş bir Gensim doc2vec modelinden bir vektör temsili elde etmek için *infer_vector()* metotu kullanılır. Bu metodun 500.000 kez yürütülmesi 4 ila 42 saat sürmektedir. Bu sürenin bu kadar geniş bir aralıkta seyretmesinin sebebi eğitim verisinin büyüklüğü, vektör boyutu ve diğer model parametrelerinden kaynaklanan etmenlerdir. Bu süre, kullanıcılarla ilgili tweet, profil bilgisi ve etkileşim verilerinin bir araya getirildiği derlemin doküman modelinin eğitimi için kullanıldığı durumlarda en uzundur.

Deneylerin koşturulmasında Fırat Üniversitesi Büyük Veri ve Yapay Zekâ Laboratuvarı hesaplama kaynaklarından olan 4 makine kullanılmıştır. Her makinede **48-core Intel(R) Xeon(R) CPU E5** ve **256GB RAM** bulunmaktadır. Makinelerden ikisi **E5-2650**, diğer ikisi ise **E5-2628L** işlemciye sahiptir. Zaman ölçümlerinde işlemci farkı önemsiz kabul edilmiştir. Veri seti büyüklüğü ve model parametrelerinin hesaplama performansını nasıl etkilediği hakkında bir fikir vermek amacıyla deneylerle ilgili çeşitli çalışma süreleri sunulmuştur.

Tablo 6.26. Ortalama hesaplama süreleri

Veri seti	İterasyon Süresi (Dk.)	Vektör çıkartım süresi (Dk.)
A	0.5	256
B	10.4	303
C	1.5	282
A + B	11.16	321
A + C	2	286
B + C	12.93	322
A + B + C	13.46	330
A + E	2.33	2152
A + B + E	11.18	2521
A + C + E	5.11	2189
A + B + C + E	13.8	2534

Gensim kütüphanesinde, iş parçacıkları (thread) yardımıyla doküman modeli eğitiminde paralelleştirme mümkündür. Bu özellik, eğitimin başlatılması sırasında modele aktarılan "$workers$" parametresiyle aktif edilir. İşlemcilerin her biri 48 çekirdeğe sahip olduğu için her biri aynı anda 48 iş parçacığı çalıştırabilir. Eğitimlerde iş parçacığı sayısı (workers) 36 olarak tercih edilmiştir. Gensim'de model eğitim süresinde yüksek performans sağlayan bu işlev ne yazık ki vektör çıkartım mekanizmasını gerçekleştiren "$infer\_vector\ ()$" yordamında tanımlı değildir. Bu çalışma kapsamında sarf edilen toplam deney sürelerinin yaklaşık olarak $\%95 - \%99$ arasında

bir bölümü doküman metinlerinden vektör temsillerinin elde edilmesi için harcanmıştır. Dolayısıyla paralelleştirme kullanımıyla ancak kısmen bir hızlanma kaydedilmiştir. Vektör çıkartım mekanizması için paralelleştirme işlevine ihtiyaç olduğu aşikârdır.

Tablo 6.26'da, yürütülme süresi bakımından deneyler arasında genel bir karşılaştırma yapılmıştır. İterasyon süresi sütunu; modelin eğitimi boyunca bir iterasyon için geçen ortalama süreyi göstermektedir. Vektör çıkartım süresi sütunu; 500.000 kayıt için geçen toplam vektör çıkartım (infer metodu çağrıları) süresini göstermektedir. Her bir satırda örnek olarak verilen deneylerde eğitim parametreleri ($dm = 0$, $min\_count = 1$) ortak olup farklı veri setleri kullanılmıştır. Parametrelerin farklı seçimlerinde sürede az miktarda değişim gözlenmiştir. Aynı deney için dm parametresi 0 yerine 1 olarak seçildiğinde iterasyon süresi artmaktadır ve "min count" parametresi arttıkça iterasyon süresi azalmaktadır.

6.8. Sonuçların Değerlendirilmesi ve Öneriler

Bu çalışmada, son teknoloji metin temsil yöntemlerinden yararlanılarak sosyal medya kullanıcıları için gerçek değerli vektörlerin öğrenilmesine odaklanılmıştır. Önerilen yaklaşımların performansını değerlendirmek için manuel olarak seçilen 500 Twitter kullanıcısının sosyal medya verileri toplanarak, literatüre türünün tek örneği bir veri seti kazandırılmıştır. Kapsamlı deneysel sonuçlar, kullanıcı temsili görevlerinde hem metinsel hem de metinsel olmayan özelliklerden yararlanmak konusuyla ilgilenen endüstri ve akademiden araştırmacılara rehberlik edecektir.

Geleneksel yöntemler (TFIDF), klasik kelime gömmeleri (word2vec, Glove, FastText vb.), doc2vec algoritmaları (PV-DBOW, PV-DM) ve bağlamsal metin temsil yöntemleri (ELMO, BERT) üzerinden ayrıntılı bir karşılaştırma sunulmuştur. Literatürde öne çıkan yöntemlerin ve bu yöntemlerin başarısına katkı sağlama potansiyeli olan farklı türden veri kaynaklarının kullanıldığı onlarca deneyin gerçekleştirilmesi sonucunda ortaya çıkan önemli sonuçlar şu şekildedir:

1. Önerilen doküman tabanlı modellerin, ek sosyal medya derlemlerinden ve kullanıcı aktivite verilerinden yararlanarak kullanıcı gömme kalitesini iyileştirebildiği gösterilmiştir. Model eğitiminde *B* ve *D* veri setlerinin kullanıldığı durumlar haricindeki tüm durumlarda PV-DM algoritması ve "min count = 5" parametre seçimleri en yüksek doğruluk değerlerini vermiştir. *B* ve *D* veri setlerine özel parametre seçimi etkisi önemsiz kabul edilebilir. Çünkü *Deney senaryoları #1* – 2 numaralı deney ve *Deney senaryoları #2* – 1 numaralı deneyde bu veri setlerinin model başarısını arttırmadığı gösterilmiştir. PV-DM algoritmasının DBOW'a göre daha yüksek başarı göstermesi sonucu, Lau ve Baldwin'in deneysel değerlendirmeleriyle [150] çelişmektedir.

2. Önerilen doc2vec modelleriyle ilgili olarak; probleme özgü veri setlerinin oluşturulması gerçek hayat uygulamaları için maliyetli bir çözümdür. *A* veri seti ve zenginleştirilmiş versiyonları olan *D* ve *E* veri setleri çoğu zaman kullanıma hazır bir şekilde mevcut olmayacaktır. Bu nedenle, *C* veri setine benzer veri setlerinin kullanılmasını öneriyoruz. Önerdiğimiz doc2vec tabanlı model *C* veri seti ile %97,5 doğru temsil başarısı sağlamıştır.

3. *C* veri seti benzeri bir veri setiyle eğitilmiş modelin daha iyi başarı göstermesi için, eğer bir odak kullanıcı grubu varsa, bu grupla ilgili kullanıcı tweetleri ve sosyal medya verilerinin (beğeni, yorum, profil bilgileri vb.) önerdiğimiz veri zenginleştirme yöntemleriyle derleme eklenmesi faydalı olacaktır. Bunun için *C* benzeri, alternatif hazır derlemler bulunabilir. Bir diğer seçenek ise; anahtar kelime, hashtag, tarih aralığı, konum, takipçi sayısı ve daha bir çok sınırlandırıcı filtreler yardımıyla veri toplamaktır.

4. PWE'ler, kullanımı çok basit olması ve kabul edilebilir başarı göstermesine rağmen, doc2vec yaklaşımları kadar iyi performans göstermemektedir. Eğer yine de sadece PWE'ler yardımıyla kullanıcı temsilleri elde edilecekse nasıl bir derlemden eğitilerek elde edildiğine ve vektör boyutu seçimine dikkat edilmelidir. *A* veri setinin derlem olarak kullanılması ile eğittimiz word2vec modeli %84,5 başarı göstermiştir. Genel bir tweet derlemi üzerinde eğitilmiş hazır Glove modeli ise %84,0 başarı gösterebilmiştir.

5. BERT ve ELMO bağlamsal metin temsil yöntemlerinin analize konu olan kullanıcılara ait tweetlerle bir iyileştirme (fine-tuning) aşamasına tabi tutulmadan, doğrudan kullanılması Glove, word2vec gibi geleneksel kelime temsil yöntemlerinin üzerinde bir başarı göstermemektedir. Başarıyı arttırmak için modelin yeni verilerle adaptasyonu veya baştan eğitilmesi seçenekleri değerlendirilmedir. Son zamanlarda bir çok DDİ probleminde kaydadeğer performans artışları sağlayan bağlamsal modeller, kullanıcı temsil görevi için de gelecek vaat etmektedir.

7. SONUÇLAR

Bu tez çalışmasında güncel olan en gelişmiş metin analitiği tekniklerini kullanarak Twitter örneği üzerinden bir sosyal ağın kullanıcıları hakkında otomatik olarak anlamlı gözlemler oluşturma görevinde kullanılabilecek, kullanıcıların vektör uzayında semantik temsillerini üreten kullanıcı temsil modelleri başarıyla geliştirilmiştir. Kullanıcı gömmelerinin etkinliği, hazırlanan altın standart bir veri seti ile test edilerek ortaya konulmuştur. Bu testlerde; doc2vec, ELMO, BERT gibi farklı algoritmaların kullanımının yanı sıra nasıl bir veri setinin tercih edilmesi gerektiği detaylı olarak irdelenmiştir. Böylelikle, kullanıcı temsil modeli gereksinimi olan gerçek dünya uygulamalarının spesifik koşullar içerisinde en iyi sonucu verecek şekilde geliştirilmesine rehberlik edecek temsil yöntemi, parametre seçimi, veri seti yapısı gibi önemli bilgiler literatüre kazandırılmıştır. Ayrıca, sosyal medya verileri üzerinde duygu analizi, metin sınıflandırma yöntemleri için derin öğrenme tabanlı yaklaşımlardan yararlanarak veri dengesizliği ve alan adaptasyonu konularında çalışmalar sunulmuştur.

Önerilen kullanıcı temsili yaklaşımı sosyal medya analizi kapsamında birçok uygulamaya katkı sağlama potansiyeline sahiptir. Bu uygulamalara aşağıdaki muhtemel senaryolar örnek verilebilir:

- **Kullanıcı benzerliği:** DEĞİRMEN Projesi kapsamında geliştirilen kullanıcı benzerlik tespiti modülünde olduğu gibi bir ağ içerisinde bir kullanıcıya en benzer kullanıcılar sorgulanabilir. Bu sistem ile bir kullanıcı, benzer diğer kullanıcılar üzerinden tanımlanabilir. Bu tanımlama; eğilim analizi, itibar analizi gibi farklı diğer sosyal medya analizi uygulamaları içerisinde faydalı bilgi olarak kullanılabilir.
- **Sosyal medya duygu analizi:** Bir metnin pozitif veya negatif olduğuna sadece daha önce etiketlenmiş etiketli diğer metinlerden öğrenerek karar veren bir model yerine her bir metnin yazarının kim olduğunun da öğrenme sürecine dâhil edilmesi sağlanabilir. Bu sayede, tıpkı gerçek hayatta olduğu gibi bir görüş dikkate alınırken genel olarak ne ifade ettiğinin yanı sıra, söyleyen kişiye bağlı olarak daha doğru anlaşılması gibi bir hedef belirlenebilir.
- **Sahte haber tespiti:** Sosyal medya üzerinde yayılmakta olan bir bilginin doğru olup olmadığının erken tespiti günümüz internet dünyasının önemli bir ihtiyacıdır. Duygu analizi örneğine benzer olarak bir paylaşımı sadece içeriğiyle değil onu oluşturan ve destek veren kullanıcıların önerdiğimiz yöntemle elde edilen temsilleriyle ele alındığı bir sistem tasarlanabilir.

TÜBİTAK tarafından desteklenen ve başarıyla tamamlanan “Sosyal Ağlar İçin Bulut Tabanlı, Dağıtık İtibar Analizi Sistemi” projesinin ürünü olan sosyal medya analizi platformu için gerçekleştirilen sosyal ağ verilerinin yönetimi (veri toplama, depolama, sorgulama), itibar analizi, duygu analizi ve çeşitli istatistiksel analizlerin araştırma ve geliştirme faaliyetlerinde tez çıktılarının önemli katkısı olmuştur.

Tez kapsamında geliştirilen çalışmaların çıktıları DEĞİRMEN projesinin sosyal medya kullanıcı benzerliği, eğilim tespiti, duygu analizi ve metin sınıflandırma modüllerinde kullanılmaktadır.

Ulusal ve uluslararası akademik yayınlar (bildiri ve makaleler) ve herkese açık olarak paylaşılan veri setleri ile literatüre katkıda bulunulmuştur.

KAYNAKLAR

[1] R. V Zicari, J. Brodersen, J. Brusseau, B. Düdder, T. Eichhorn, T. Ivanov, G. Kararigas, P. Kringen, M. McCullough, F. Möslein, others, Z-Inspection®: A Process to Assess Trustworthy AI, IEEE Trans. Technol. Soc. (2021).

[2] A. Rajaraman, J.D. Ullman, Mining of massive datasets, Cambridge University Press, 2011.

[3] G. Adomavicius, R. Sankaranarayanan, S. Sen, A. Tuzhilin, Incorporating contextual information in recommender systems using a multidimensional approach, ACM Trans. Inf. Syst. 23 (2005) 103–145.

[4] K. Żołna, B. Romański, User modeling using LSTM networks, in: Thirty-First AAAI Conf. Artif. Intell., 2017.

[5] S. Pan, T. Ding, Social media-based user embedding: a literature review, ArXiv Prepr. ArXiv1907.00725. (2019).

[6] M. Kaya, Sosyal Medya ve Sosyal Medyada Üçüncü Kişilerin Kişilik Haklarının İhlali, Türkiye Barolar Birliği Derg. (2015) 277–306.

[7] C.E. Shannon, A Mathematical Theory of Communication, Bell Syst. Tech. J. (1948). https://doi.org/10.1002/j.1538-7305.1948.tb01338.x.

[8] R.E. Krebs, C.A. Krebs, Groundbreaking scientific experiments, inventions, and discoveries of the ancient world, Greenwood Publishing Group, 2003.

[9] K.S. Jones, Natural language processing: a historical review, Univ. Cambridge. (2001) 2–10.

[10] C. Bally, A. Sechehaye, Course in general linguistics Ferdinand de Saussure, McGraw-Hill Book Company, 1966.

[11] J. Hutchins, Machine translation: A concise history, Comput. Aided Transl. Theory Pract. 13 (2007) 11.

[12] J. Hutchins, E. Lovtskii, Petr Petrovich Troyanskii (1894--1950): A forgotten pioneer of mechanical translation, Mach. Transl. 15 (2000) 187–221.

[13] N.R.C. (US). A.L.P.A. Committee, Language and machines: computers in translation and linguistics; a report, National Academies, 1966.

[14] M. Sanderson, W.B. Croft, The history of information retrieval research, Proc. IEEE. 100 (2012) 1444–1451.

[15] P. Sheridan, Research in language translation on the IBM type 701, IBM Tech. Newsl. 9 (1955) 5–24.

[16] A.M. Turing, I.—Computing Machinery and Intelligence, Mind. LIX (1950) 433–460. https://doi.org/10.1093/mind/LIX.236.433.

[17] J. McCarthy, M.L. Minsky, N. Rochester, C.E. Shannon, A proposal for the dartmouth summer research project on artificial intelligence, august 31, 1955, AI Mag. 27 (2006) 12.

[18] J. Mccarthy, Programs with Common Sense, in: Mech. Thought Process. Proc. Symp. Natl. Phys. Lab., London, U.K., 1959: pp. 77–84.

[19] R.B. Lees, N. Chomsky, Syntactic Structures, Language (Baltim). 33 (1957) 375. https://doi.org/10.2307/411160.

[20] T. Winograd, Understanding natural language, Cogn. Psychol. 3 (1972) 1–191.

[21] F.B. Baker, Latent class analysis as an association model for information retrieval, Stat. Assoc. Methods Mech. Doc. Natl. Bur. Stand. Misc. Publ. (1965) 149–155.

[22] J. Spiegel, E. Bennett, A Modified Statistical Association Procedure for Automatic Document Content Analysis and Retrieval, in: Stat. Assoc. Methods Mech. Doc. Symp. Proc., 1965: p. 47.

[23] H.P. Luhn, A statistical approach to mechanized encoding and searching of literary information, IBM J. Res. Dev. 1 (1957) 309–317.

[24] Y. Bengio, R. Ducharme, P. Vincent, C. Jauvin, A neural probabilistic language model, J. Mach. Learn. Res. 3 (2003) 1137–1155.

[25] Z.S. Harris, Distributional Structure, *WORD*. (1954). https://doi.org/10.1080/00437956.1954.11659520.

[26] J.R. Firth, A synopsis of linguistic theory, 1930-1955, Stud. Linguist. Anal. (1957).

[27] K.S. Jones, A statistical interpretation of term specificity and its application in retrieval, J. Doc. (1972).

[28] G. Salton, C. Buckley, Term-weighting approaches in automatic text retrieval, Inf. Process. Manag. 24 (1988) 513–523.

[29] J. Beel, S. Langer, B. Gipp, Tf-iduf: A novel term-weighting scheme for user modeling based on users' personal document collections, IConference 2017 Proc. (2017).

[30] G. Salton, A. Wong, C.-S. Yang, A vector space model for automatic indexing, Commun. ACM. 18 (1975) 613–620.
[31] S. Deerwester, S. Dumais, T. Landauer, G. Furnas, L. Beck, Improving information-retrieval with latent semantic indexing, in: Proc. ASIS Annu. Meet., 1988: pp. 36–40.
[32] S.T. Dumais, Latent semantic analysis, Annu. Rev. Inf. Sci. Technol. 38 (2004) 189–230.
[33] L. Kovács, P. Barabás, Grammar Representation Forms in Natural Language Interface for Robot Controlling, in: Emergent Trends Robot. Intell. Syst., Springer, 2015: pp. 65–72.
[34] M.D. Harris, Introduction to natural language processing, Reston Publishing Co., 1985.
[35] G.A. Miller, WordNet: a lexical database for English, Commun. ACM. 38 (1995) 39–41.
[36] O. Bilgin, Ö. Çetinoğlu, K. Oflazer, Building a Wordnet for Turkish, Rom. J. Inf. Sci. Technol. 7 (2004) 163–172.
[37] A. Akbik, T. Bergmann, D. Blythe, K. Rasul, S. Schweter, R. Vollgraf, FLAIR: An Easy-to-Use Framework for State-of-the-Art NLP, in: Proc. 2019 Conf. North, 2019. https://doi.org/10.18653/v1/N19-4010.
[38] M. Wettler, R. Rapp, Computation of Word Associations Based on Co-occurrences of Words in Large Corpora, in: Very Large Corpora Acad. Ind. Perspect., 1993. https://www.aclweb.org/anthology/W93-0310.
[39] M.F. Amasyali, A. Beken, Measurement of Turkish word semantic similarity and text categorization application, in: 2009 IEEE 17th Signal Process. Commun. Appl. Conf., 2009: pp. 1–4.
[40] E. Arisoy, M. Saraçlar, Language modeling for Turkish text and speech processing, in: Turkish Nat. Lang. Process., Springer, 2018: pp. 69–92.
[41] F. Jelinek, R.L. Mercer, L.R. Bahl, J.K. Baker, Perplexity—a measure of the difficulty of speech recognition tasks, J. Acoust. Soc. Am. 62 (1977) S63--S63.
[42] P.F. Brown, V.J. Della Pietra, P. V Desouza, J.C. Lai, R.L. Mercer, Class-based n-gram models of natural language, Comput. Linguist. 18 (1992) 467–480.
[43] A.L. Samuel, Some studies in machine learning using the game of checkers, IBM J. Res. Dev. 3 (1959) 210–229.
[44] C.M. Bishop, Pattern recognition and machine learning, springer, 2006.
[45] D. Servan-Schreiber, A. Cleeremans, J.L. McClelland, Learning sequential structure in simple recurrent networks, in: Adv. Neural Inf. Process. Syst., 1989: pp. 643–652.
[46] J.L. Elman, Finding structure in time, Cogn. Sci. 14 (1990) 179–211.
[47] J.L. Elman, Distributed representations, simple recurrent networks, and grammatical structure, Mach. Learn. 7 (1991) 195–225.
[48] P. Munro, C. Cosic, M. Tabasko, A network for encoding, decoding and translating locative prepositions, Conn. Sci. 3 (1991) 225–240.
[49] H. Schütze, Part-of-speech induction from scratch, in: 31st Annu. Meet. Assoc. Comput. Linguist., 1993: pp. 251–258.
[50] N.K. Imperial, N. Koncar, D.G. Guthrie, A Natural Language Translation Neural Network, in: Proc. Int. Conf. New Methods Lang. Process. (Neml., 1994: pp. 71–77.
[51] R. Miikkulainen, Subsymbolic case-role analysis of sentences with embedded clauses, Cogn. Sci. 20 (1996) 47–73.
[52] C. Manning, H. Schutze, Foundations of statistical natural language processing, MIT press, 1999.
[53] D.E. Rumelhart, J.L. McClelland, On learning the past tenses of English verbs, (1986).
[54] W. Xu, A. Rudnicky, Can artificial neural networks learn language models?, in: Sixth Int. Conf. Spok. Lang. Process., 2000.
[55] H. Robbins, S. Monro, A stochastic approximation method, Ann. Math. Stat. (1951) 400–407.
[56] H. Schwenk, D. Déchelotte, J.-L. Gauvain, Continuous space language models for statistical machine translation, in: Proc. COLING/ACL 2006 Main Conf. Poster Sess., 2006: pp. 723–730.
[57] J. Turian, L. Ratinov, Y. Bengio, Word representations: a simple and general method for semi-supervised learning, in: Proc. 48th Annu. Meet. Assoc. Comput. Linguist., 2010: pp. 384–394.
[58] G.E. Hinton, Distributed representations, (1984).
[59] S. Lai, K. Liu, S. He, J. Zhao, How to generate a good word embedding, IEEE Intell. Syst. 31 (2016) 5–14.
[60] R. Collobert, J. Weston, A unified architecture for natural language processing: Deep neural networks with multitask learning, in: Proc. 25th Int. Conf. Mach. Learn., 2008: pp. 160–167.
[61] T. Mikolov, W. Yih, G. Zweig, Linguistic regularities in continuous space word representations, in: Proc. 2013 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol., 2013: pp. 746–751.

[62] D. Jurgens, S. Mohammad, P. Turney, K. Holyoak, Semeval-2012 task 2: Measuring degrees of relational similarity, in: * SEM 2012 First Jt. Conf. Lex. Comput. Semant. 1 Proc. Main Conf. Shar. Task, Vol. 2 Proc. Sixth Int. Work. Semant. Eval. (SemEval 2012), 2012: pp. 356–364.
[63] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, ArXiv Prepr. ArXiv1301.3781. (2013).
[64] T. ve diğerleri Mikolov, Word2vec Projesi, Word2Vec Proj. (2014). https://code.google.com/p/word2vec (accessed August 18, 2020).
[65] A. Aravkin, A. Choromanska, L. Deng, G. Heigold, T. Jebara, D. Kanevsky, S.J. Wright, Log-linear Models, Extensions, and Applications, MIT Press, 2018.
[66] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: Adv. Neural Inf. Process. Syst., 2013: pp. 3111–3119.
[67] J. Pennington, R. Socher, C.D. Manning, Glove: Global Vectors for Word Representation., in: EMNLP, 2014: pp. 1532–1543.
[68] P. Bojanowski, E. Grave, A. Joulin, T. Mikolov, Enriching Word Vectors with Subword Information, Trans. Assoc. Comput. Linguist. (2017). https://doi.org/10.1162/tacl_a_00051.
[69] R. Speer, J. Chin, C. Havasi, Conceptnet 5.5: An open multilingual graph of general knowledge, in: Thirty-First AAAI Conf. Artif. Intell., 2017.
[70] R. Speer, J. Lowry-Duda, ConceptNet at SemEval-2017 Task 2: Extending Word Embeddings with Multilingual Relational Knowledge, in: 2018. https://doi.org/10.18653/v1/s17-2008.
[71] M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, L. Zettlemoyer, Deep Contextualized Word Representations, ArXiv Prepr. ArXiv1802.05365. (2018). https://doi.org/10.18653/v1/n18-1202.
[72] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, ArXiv Prepr. ArXiv1810.04805. (2018).
[73] H. Öztürk, A. Degirmenci, O. Güngör, S. Uskudarli, The Role of Contextual Word Embeddings in Correcting the 'de/da'Clitic Errors in Turkish, in: 2020 28th Signal Process. Commun. Appl. Conf., 2020: pp. 1–4.
[74] U.U. Acikalin, B. Bardak, M. Kutlu, Turkish Sentiment Analysis Using BERT, in: 2020 28th Signal Process. Commun. Appl. Conf., 2020: pp. 1–4.
[75] B. Pang, L. Lee, Opinion mining and sentiment analysis, Found. Trends Inf. Retr. (2008). https://doi.org/10.1561/1500000011.
[76] B. Liu, Sentiment analysis and opinion mining, Synth. Lect. Hum. Lang. Technol. 5 (2012) 1–167.
[77] W. Medhat, A. Hassan, H. Korashy, Sentiment analysis algorithms and applications: A survey, Ain Shams Eng. J. 5 (2014) 1093–1113.
[78] A. Alpkoçak, M.A. Tocoglu, A. Çelikten, I. Aygün, Türkçe Metinlerde Duygu Analizi için Farklı Makine Öğrenmesi Yöntemlerinin Karşılaştırılması, Dokuz Eylül Üniversitesi Mühendislik Fakültesi Fen ve Mühendislik Derg. 21 (2019) 719–725.
[79] E. Grave, P. Bojanowski, P. Gupta, A. Joulin, T. Mikolov, Learning Word Vectors for 157 Languages, in: Proc. Int. Conf. Lang. Resour. Eval. (LREC 2018), 2018.
[80] L. Deng, D. Yu, others, Deep learning: methods and applications, Found. Trends® in Signal Process. 7 (2014) 197–387.
[81] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016: pp. 770–778.
[82] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Adv. Neural Inf. Process. Syst., 2012: pp. 1097–1105.
[83] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, ArXiv Prepr. ArXiv1409.1556. (2014).
[84] F. Chollet, Keras: The Python Deep Learning Library, (2015). Keras.io (accessed January 20, 2020).
[85] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D.G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, X. Zheng, TensorFlow: A system for large-scale machine learning, in: Proc. 12th USENIX Symp. Oper. Syst. Des. Implementation, OSDI 2016, 2016.
[86] G. Hernández, E. Zamora, H. Sossa, G. Téllez, F. Furlán, Hybrid neural networks for big data classification, Neurocomputing. 390 (2020) 327–340.
[87] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE. 86 (1998) 2278–2324.
[88] T. Young, D. Hazarika, S. Poria, E. Cambria, Recent trends in deep learning based natural language

processing, ArXiv Prepr. ArXiv1708.02709. (2017).
[89] X. Shi, Z. Chen, H. Wang, D.Y. Yeung, W.K. Wong, W.C. Woo, Convolutional LSTM network: A machine learning approach for precipitation nowcasting, Adv. Neural Inf. Process. Syst. 2015 (2015) 802–810.
[90] S. Bai, J.Z. Kolter, V. Koltun, An empirical evaluation of generic convolutional and recurrent networks for sequence modeling, ArXiv Prepr. ArXiv1803.01271. (2018).
[91] J. Yoon, H. Kim, Multi-channel lexicon integrated CNN-BiLSTM models for sentiment analysis, in: Proc. 29th Conf. Comput. Linguist. Speech Process. (ROCLING 2017), 2017: pp. 244–253.
[92] T. Elsken, J.H. Metzen, F. Hutter, others, Neural architecture search: A survey., J. Mach. Learn. Res. 20 (2019) 1–21.
[93] P. Liashchynskyi, P. Liashchynskyi, Grid search, random search, genetic algorithm: A big comparison for nas, ArXiv Prepr. ArXiv1912.06059. (2019).
[94] J. Bergstra, Y. Bengio, Random search for hyper-parameter optimization., J. Mach. Learn. Res. 13 (2012).
[95] Y. Bengio, I. Goodfellow, A. Courville, Deep learning, MIT press Massachusetts, USA:, 2017.
[96] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: A simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (2014) 1929–1958.
[97] L. Datta, A survey on activation functions and their relation with xavier and he normal initialization, ArXiv Prepr. ArXiv2004.06632. (2020).
[98] L. Luo, Y. Xiong, Y. Liu, X. Sun, Adaptive gradient methods with dynamic bound of learning rate, ArXiv Prepr. ArXiv1902.09843. (2019).
[99] T. Tieleman, G. Hinton, Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude, COURSERA Neural Networks Mach. Learn. 4 (2012) 26–31.
[100] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, ArXiv Prepr. ArXiv1412.6980. (2014).
[101] S. Cateni, V. Colla, M. Vannucci, A method for resampling imbalanced datasets in binary classification tasks for real-world problems, Neurocomputing. 135 (2014) 32–41.
[102] N.H. (https://stats.stackexchange.com/users/76825/nuclear-hoagie), Am I allowed to average the list of precision and recall after k-fold cross validation?, (2019). https://stats.stackexchange.com/q/388944 (accessed August 18, 2020).
[103] M. Imran, P. Mitra, C. Castillo, Twitter as a lifeline: Human-annotated twitter corpora for NLP of crisis-related messages, ArXiv Prepr. ArXiv1605.05894. (2016).
[104] I.R. Hallac, B. Ay, G. Aydin, Experiments on Fine Tuning Deep Learning Models With News Data For Tweet Classification, in: 2018 Int. Conf. Artif. Intell. Data Process. IDAP 2018, 2019. https://doi.org/10.1109/IDAP.2018.8620869.
[105] S. Bai, J.Z. Kolter, V. Koltun, An empirical evaluation of generic convolutional and recurrent networks for sequence modeling, ArXiv. (2018).
[106] Y. Kim, Convolutional neural networks for sentence classification, ArXiv Prepr. ArXiv1408.5882. (2014).
[107] K. Cao, M. Rei, A joint model for word embedding and word morphology, ArXiv Prepr. ArXiv1606.02601. (2016).
[108] R. Vargas, A. Mosavi, R. Ruiz, Deep learning: a review, Adv. Intell. Syst. Comput. (2017).
[109] G. Aydin, RssTurk, (2018). https://github.com/galipaydin/rssturk (accessed February 1, 2018).
[110] A. Joulin, E. Grave, P. Bojanowski, T. Mikolov, Bag of Tricks for Efficient Text Classification, in: Proc. 15th Conf. Eur. Chapter Assoc. Comput. Linguist. Vol. 2, Short Pap., Association for Computational Linguistics, 2017: pp. 427–431.
[111] A. Marakis, BiLSTM-CNN Model for Toxic Comment Classification Challenge, (n.d.). https://www.kaggle.com/antmarakis/bi-lstm-conv-layer (accessed January 15, 2020).
[112] J.P.C. Chiu, E. Nichols, Named entity recognition with bidirectional LSTM-CNNs, Trans. Assoc. Comput. Linguist. 4 (2016) 357–370.
[113] B. Ay Karakus, M. Talo, I.R. Hallaç, G. Aydin, Evaluating deep learning models for sentiment classification, Concurr. Comput. Pract. Exp. 30 (2018) e4783.
[114] T. Van Huynh, V.D. Nguyen, K. Van Nguyen, N.L.-T. Nguyen, A.G.-T. Nguyen, Hate Speech Detection on Vietnamese Social Media Text using the Bi-GRU-LSTM-CNN Model, ArXiv Prepr. ArXiv1911.03644. (2019).
[115] D.P. Kingma, J.L. Ba, Adam: A method for stochastic optimization, in: 3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc., 2015.
[116] I.R. Hallac, S. Makinist, B. Ay, G. Aydin, user2Vec: Social Media User Representation Based on Distributed Document Embeddings, in: 2019 Int. Artif. Intell. Data Process. Symp., 2019: pp. 1–5.

[117] J.A. Feldman, P.J. Hayes, D.E. Rumelhart, Parallel Distributed Processing computational model of cognition and perception, 1986. https://doi.org/https://www.indracompany.com.
[118] Q. V. Le, T. Mikolov, Distributed Representations of Sentences and Documents, 32 (2014). https://doi.org/10.1145/2740908.2742760.
[119] L. Chen, T. Qian, P. Zhu, Z. You, Learning user embedding representation for gender prediction, in: 2016 IEEE 28th Int. Conf. Tools with Artif. Intell., 2016: pp. 263–269.
[120] A. Lay, B. Ferwerda, Predicting users' personality based on their 'liked'images on instagram, in: 23rd Int. Intell. User Interfaces, March 7-11, 2018, 2018.
[121] F. Mairesse, M.A. Walker, M.R. Mehl, R.K. Moore, Using linguistic cues for the automatic recognition of personality in conversation and text, J. Artif. Intell. Res. 30 (2007) 457–500.
[122] S. Wooders, R. Senanayake, FashionModel : Mapping Images of Clothes to an Embedding Space, (2014) 2–5.
[123] E. Asgari, M.R.K. Mofrad, Continuous distributed representation of biological sequences for deep proteomics and genomics, PLoS One. 10 (2015) 1–15. https://doi.org/10.1371/journal.pone.0141287.
[124] A. Viehweger, S. Krautwurst, B. Koenig, M. Marz, Distributed representations of protein domains and genomes and their compositionality, BioRxiv. (2019) 524280. https://doi.org/10.1101/524280.
[125] P. Covington, J. Adams, E. Sargin, Deep neural networks for youtube recommendations, in: Proc. 10th ACM Conf. Recomm. Syst., 2016: pp. 191–198. https://doi.org/10.1145/3132847.3133154.
[126] B. Esmaeili, H. Huang, B.C. Wallace, J.-W. van de Meent, Structured Representations for Reviews:Aspect-Based Variational Hidden Factor Models, 89 (2018). https://arxiv.org/abs/1812.05035.
[127] P. Mishra, M. Del Tredici, H. Yannakoudakis, E. Shutova, Author Profiling for Abuse Detection, in: COLING, 2018.
[128] Y. Yu, X. Wan, X. Zhou, User Embedding for Scholarly Microblog Recommendation, in: Proc. 54th Annu. Meet. Assoc. Comput. Linguist., 2016: pp. 449–453.
[129] S. Amir, G. Coppersmith, P. Carvalho, M.J. Silva, B.C. Wallace, Quantifying Mental Health from Social Media with Neural User Embeddings, 2008 (2017) 1–17. https://doi.org/10.11607/jomi.3185.
[130] A. Jaech, S. Hathi, M. Ostendorf, Community Member Retrieval on Social Media using Textual Information, Naacl. (2018) 595–601. http://aclweb.org/anthology/N18-2094%0Ahttp://arxiv.org/abs/1804.05499.
[131] A. Benton, M. Dredze, Using Author Embeddings to Improve Tweet Stance Classification, (2018) 184–194.
[132] L. Xing, M.J. Paul, Incorporating Metadata into Content-Based User Embeddings, Proc. 3rd Work. Noisy User-Generated Text. (2017) 45–49. http://aclweb.org/anthology/W17-4406.
[133] S. Amir, B.C. Wallace, H. Lyu, P.C.M.J. Silva, Modelling context with user embeddings for sarcasm detection in social media, ArXiv Prepr. ArXiv1607.00976. (2016). https://doi.org/10.18653/v1/K16-1017.
[134] R. Rehurek, P. Sojka, others, Gensim—statistical semantics in python, Retrieved from Genism. Org. (2011).
[135] E. Loper, S. Bird, NLTK: the natural language toolkit, ArXiv Prepr. Cs/0205028. (2002).
[136] I.R. Hallac, B. Ay, G. Aydin, User Representation Learning for Social Networks: An Empirical Study, Appl. Sci. 11 (2021) 5489.
[137] J. Zhao, F. Van Harmelen, J. Tang, X. Han, Q. Wang, X. Li, Knowledge Graph and Semantic Computing. Knowledge Computing and Language Understanding: Third China Conference, CCKS 2018, Tianjin, China, August 14--17, 2018, Revised Selected Papers, Springer, 2018.
[138] J. Littman, L. Wrubel, D. Kerchner, Y. Bromberg Gaber, News Outlet Tweet Ids, (2017). https://doi.org/10.7910/DVN/2FIFLH.
[139] P. Binkley, twarc-report README. md, (2015). https://github.com/DocNow/twarc (accessed February 20, 2021).
[140] J. Roesslein, tweepy Documentation, Online] Http://Tweepy. Readthedocs. Io/En/V3. 5 (2009).
[141] Y. Yamamoto, Twitter4j-Twitter API için Java Kütüphanesi, (2010). http://twitter4j.org/en/.
[142] Bear - A Python wrapper around the Twitter API., (n.d.). https://github.com/bear/python-twitter (accessed February 20, 2021).
[143] L. Richardson, Beautiful soup documentation, Dosegljivo Https//Www. Crummy. Com/Software/BeautifulSoup/Bs4/Doc/.[Dostopano 7. 7. 2018]. (2007).
[144] P. Jaccard, Nouvelles recherches sur la distribution florale, Bull. Soc. Vaud. Sci. Nat. 44 (1908) 223–270.

[145] M.K. Vijaymeena, K. Kavitha, A survey on similarity measures in text mining, Mach. Learn. Appl. An Int. J. 3 (2016) 19–28.
[146] P.D. Hoff, A.E. Raftery, M.S. Handcock, Latent space approaches to social network analysis, J. Am. Stat. Assoc. (2002). https://doi.org/10.1198/016214502388618906.
[147] A.M. Dai, C. Olah, Q. V Le, Document embedding with paragraph vectors, ArXiv Prepr. ArXiv1507.07998. (2015).
[148] J. Pennington, S. Richard, M. Christopher D., Glove: pre-trained word vectors download page, (2014). https://nlp.stanford.edu/projects/glove/.
[149] L. van der Maaten, G. Hinton, Visualizing data using t-SNE, J. Mach. Learn. Res. 9 (2008) 2579–2605.
[150] J.H. Lau, T. Baldwin, An empirical evaluation of doc2vec with practical insights into document embedding generation, ArXiv Prepr. ArXiv1607.05368. (2016).

ÖZGEÇMİŞ

İbrahim Rıza HALLAÇ

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

DİĞER AKADEMİK DENEYİMLER

2013 : YÖK Bursu - Misafir araştırmacı (3 ay),
Indiana University, Community Grids Laboratory, Indiana, Amerika Birleşik Devletleri
Konu: "Investigating the Apache Big Data Stack", **Danışman**: Prof. Geoffrey C. Fox

2017 : Misafir araştırmacı (3 ay),
LMU - The Center for Information and Language Processing, Münih, Almanya
Konu: "Neural Machine Translation", **Danışman**: Prof. Hinrich Schütze

AKADEMİK FAALİYETLER

Makaleler:

1. I.R. Hallac, B.Ay, G.Aydin. "User Representation Learning for Social Networks: An Empirical Study" Appl. Sci. 11, no. 12 (2021): 5489.
2. B. Ay, M. Talo, I.R. Hallac, G.Aydin. "Evaluating deep learning models for sentiment classification." Concurrency and Computation: Practice and Experience 30, no. 21 (2018): e4783.
3. G. Aydin, I.R. Hallac, B. Karakus. "Architecture and implementation of a scalable sensor data storage and analysis system using cloud computing and big data technologies." Journal of Sensors 2015 (2015).
4. G. Aydin, I.R. Hallac. "Distributed log analysis on the cloud using MapReduce." Tehnički vjesnik 23, no. 4 (2016): 1011-1016.

Bildiriler:

1. I.R. Hallac, S. Makinist, B. Ay, G. Aydin. "user2Vec: Social Media User Representation Based on Distributed Document Embeddings." In 2019 International Conference on Artificial Intelligence and Data Processing (IDAP), Malatya/TURKEY, IEEE, 2019.
2. I.R. Hallac, B. Ay, G. Aydin. "Experiments on Fine Tuning Deep Learning Models With News Data For Tweet Classification." In 2018 International Conference on Artificial Intelligence and Data Processing (IDAP), Malatya/TURKEY, IEEE, 2018.
3. S. Makinist, I.R. Hallac, B.A. Karakus, G. Aydın. "Preparation of Improved Turkish DataSet for Sentiment Analysis in Social Media." In ITM Web of Conferences, vol. 13, p. 01030. EDP Sciences, Istanbul/TURKEY, 2017.
4. A.A. Albabawat, I.R. Hallac, G. Aydin. "Sentiment Analysis On A Distributed Platform." International Engineering, Science and Education Conference (INESEG), Diyarbakir/TURKEY, 2016.
5. G. Yildirim, I.R. Hallac, G. Aydin, Y. Tatar. "Running genetic algorithms on Hadoop for solving high dimensional optimization problems." In 9th International Conference on Application of Information and Communication Technologies (AICT), Rostov/RUSSIA, pp. 12-16. IEEE, 2015.
6. G. Aydin and I.R. Hallac. "Document Classification Using Distributed Machine Learning." In Second International Conference On Advances In Information Processing And Communication Technology (IPCT), Rome/ITALY, 2015.
7. G. Aydin and I.R. Hallac. "Distributed NLP." In Third International Symposium on Innovative Technologies in Engineering and Science (ISITES), Valencia/SPAIN, 2015.
8. B. Ay, I.R. Hallac, G. Aydin. "Distributed Readability Analysis of Turkish ElementarySchool Textbooks." Proceedings of International Conference on Information Technology and Computer Science, Bangkok/THAILAND, 79-87, 2015.
9. B.A. Karakus, I.R. Hallac, M. Talo, G. Aydın. "Derin Öğrenme ile Türkçe Film Yorumlarının Duygu Sınıflandırması", 5. Ulusal Yüksek Başarımlı Hesaplama Konferansı, İstanbul/TÜRKİYE, 2017.
10. S. Makinist, B.A. Karakus, I.R Hallac, M. Talo, G. Aydın. "Gensim ve Simserver ile Türkçe Haber Dokümanları Arasında Benzer Dokümanların Sınıflandırılması" (Poster), 5. Ulusal Yüksek Başarımlı Hesaplama Konferansı, İstanbul/TÜRKİYE, 2017.

Projeler:

1. Yürütücü, "**Sosyal Ağlar İçin Bulut Tabanlı, Dağıtık İtibar Analizi Sistemi**", TÜBİTAK - 1512 Teknogirişim Sermayesi Desteği Programı, 2014 – 2015.
2. Yürütücü, "**Derin Öğrenme Tabanlı Ve Yüksek Doğruluklu Yüz Tanıma İle Geçiş Yetkilendirme Ve Takip Sistemi**", KOSGEB - Ar-Ge ve İnovasyon Destek Programı, 2017 – 2019.
3. Araştırmacı, "**Derin Öğrenme Büyük Veri Analizi Platformu (DEĞİRMEN)**", Savunma Sanayii Başkanlığı- Savunma Sanayii Ar-Ge Geniş Alan Çağrısı (SAGA), 2017 – 2020.