KELİME GÖMME VEKTÖRLERİNİN GRAF DÖNÜŞÜMÜ YOLUYLA METİN SINIFLANDIRMADA KULLANIMI

2023
YÜKSEK LİSANS TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ

Elif DORUKBAŞI

Tez Danışmanı
Doç. Dr. İlker TÜRKER

KELİME GÖMME VEKTÖRLERİNİN GRAF DÖNÜŞÜMÜ YOLUYLA METİN SINIFLANDIRMADA KULLANIMI

Elif DORUKBAŞI

Tez Danışmanı
Doç. Dr. İlker TÜRKER

T.C.
Karabük Üniversitesi
Lisansüstü Eğitim Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalında
Yüksek Lisans Tezi
Olarak Hazırlanmıştır

KARABÜK
Temmuz 2023

Elif DORUKBAŞI tarafından hazırlanan "KELİME GÖMME VEKTÖRLERİNİN GRAF DÖNÜŞÜMÜ YOLUYLA METİN SINIFLANDIRMADA KULLANIMI" başlıklı bu tezin Yüksek Lisans Tezi olarak uygun olduğunu onaylarım.

Doç. Dr. İlker TÜRKER .....
Tez Danışmanı, Bilgisayar Mühendisliği Anabilim Dalı

Bu çalışma, jürimiz tarafından Oy Birliği ile Bilgisayar Mühendisliği Anabilim Dalında Yüksek Lisans tezi olarak kabul edilmiştir. 27/07/2023

Ünvanı, Adı SOYADI (Kurumu)	İmzası
Başkan : Doç. Dr. Burhan SELÇUK (KBÜ)	.....
Üye : Doç. Dr. İlker TÜRKER (KBÜ)	.....
Üye : Prof. Dr. Ergin YILMAZ (BEÜ)	ONLİNE

KBÜ Lisansüstü Eğitim Enstitüsü Yönetim Kurulu, bu tez ile, Yüksek Lisans derecesini onamıştır.

Prof. Dr. Müslüm KUZU .....
Lisansüstü Eğitim Enstitüsü Müdürü

"Bu tezdeki tüm bilgilerin akademik kurallara ve etik ilkelere uygun olarak elde edildiğini ve sunulduğunu; ayrıca bu kuralların ve ilkelerin gerektirdiği şekilde, bu çalışmadan kaynaklanmayan bütün atıfları yaptığımı beyan ederim."

Elif DORUKBAŞI

ÖZET

Yüksek Lisans Tezi

KELİME GÖMME VEKTÖRLERİNİN GRAF DÖNÜŞÜMÜ YOLUYLA METİN SINIFLANDIRMADA KULLANIMI

Elif DORUKBAŞI

Karabük Üniversitesi
Lisansüstü Eğitim Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Tez Danışmanı:
Doç. Dr. İlker TÜRKER

Temmuz 2023, 60 sayfa

Metin sınıflandırma, dijital ortamda sürekli artan miktardaki metin tabanlı belgelerin otomatik sınıflandırılması için yapay zekânın önemli bir uygulama şekli olarak karşımıza çıkmaktadır. Ulaşılabilir verinin üssel biçimde artması, metinleri hızlı biçimde işlemeye olan ihtiyaç, bilgisayarların depolama ve işlem gücünün artması, makine öğrenmesi yöntemlerindeki gelişmeler, bu alanın popülerliğini destekleyen faktörler olarak öne çıkmaktadır. Araştırmacılar metin sınıflandırması için birçok makine öğrenimi yaklaşımı ile doğal dil işlemede üstün sonuçlar elde etmiştir. Bu yaklaşımların başarısı, karmaşık modelleri ve veriler içindeki doğrusal olmayan ilişkileri anlama kapasitelerine bağlıdır. Bu noktada, graf tabanlı yaklaşımlar son yıllarda tercih edilen yöntemler arasında yer almaya başlamıştır. Öte yandan metin gömme (embedding) tekniklerindeki gelişmeler, kelimelerin anlam yükünü taşıyan vektörlerle ifade edilmesini, dolayısıyla yakın anlamlı kelimelerin de benzer

sınıflandırma sonuçlarını doğurmasını sağlamış, metin sınıflandırmada önemli bir çığır açmıştır.

Bu tez çalışmasında, literatürde sıkça kullanılan metin gömme teknikleri olan Word2Vec, GloVe, FastText ve BERT algoritmaları kullanılarak, değişken öznitelik sayısı altında yapay sinir ağları (YSA) ve derin öğrenme yöntemleri ile metin sınıflandırma yapılmış, ideal metin gömme tekniği ve öznitelik sayısının tespiti sağlanmıştır. Öznitelik belirlenmesinde, ki-kare ağırlık yönteminden yararlanılmıştır. Aynı zamanda sınıflandırma aşaması öncesinde dokümanları temsil eden vektörler görünürlük grafları (visibility graph) yaklaşımı ile graf temsillerine dönüştürülerek evrişimli sinir ağı (CNN) ile sınıflandırılmış, graf tabanlı temsillerin başarısı test edilmiştir. 2 boyutlu graf yapısı kullanılarak CNN ile karşılaştırılan bu model, diğer geleneksel yöntemlere göre daha başarılı olduğu gözlemlenmiştir. Geleneksel yöntemler ve oluşturulan graf temsilli öğrenme yaklaşımı arasında şeffaf bir karşılaştırma yapabilmek için grafları ifade eden bağlantı matrisleri tek boyuta indirgenerek YSA yöntemi ile sınıflandırma yapılmış olup %91.2 oranında bir hassasiyet elde edilmiştir.

Sonuçlar, graf temsilli yaklaşımın, geleneksel metin gömme teknikleri ile karşılaştırıldığında daha başarılı olduğunu göstermektedir. Geleneksel yöntemler arasında ise BERT'in diğer yöntemlere göre daha iyi performans gösterdiğini, FastText'in 500 kelimeye kadarki öznitelik sayıları için BERT'e yakın sonuçlar verdiğini, GloVe'un ise en düşük sınıflandırma performansı ile rekabetçi olmaktan uzak olduğunu ortaya koymaktadır. Bu tez çalışması, literatürde değişken öznitelik koşulu altında kelime gömme vektörlerinin görünürlük grafına dönüştürüldüğü ilk çalışma olarak öne çıkmaktadır.

Anahtar Sözcükler : Graf temsilli öğrenme, metin sınıflandırma, derin öğrenme, doğal dil işleme

Bilim Kodu **:** 92408

ABSTRACT

M. Sc. Thesis

USE OF WORD EMBEDDING VECTORS IN TEXT CLASSIFICATION THROUGH GRAPH CONVERSION

Elif DORUKBAŞI

Karabük University
Institute of Graduate Programs
Department of Computer Enginering

Thesis Advisor:
Assoc. Prof. Dr. İlker TÜRKER
July 2023, 60 pages

Text classification emerges as an important application form of artificial intelligence for the automatic classification of the ever-increasing amount of text-based documents in the digital environment. The exponential increase in accessible data, the need to process texts quickly, the increase in the storage and processing power of computers, and the developments in machine learning methods stand out as the factors supporting the popularity of this field. Researchers have achieved superior results in natural language processing with many machine learning approaches for text classification. The success of these approaches depends on their capacity to understand complex models and nonlinear relationships within data. At this point, graph-based approaches have started to be among the preferred methods in recent years. On the other hand, the developments in text embedding techniques have enabled words to be expressed with vectors that carry semantic load, thus causing similar classification results for words with similar meanings, breaking new ground in text classification. In this thesis, using

Word2Vec, GloVe, FastText and BERT algorithms, which are frequently used text embedding techniques in the literature, text classification was made under variable feature count with artificial neural networks (ANN) and deep learning methods, ideal text embedding technique and the number of features were determined. Chi-square weight method was used for feature determination. At the same time, before the classification stage, vectors representing documents were converted into graph representations with the visibility graph approach and classified with a convolutional neural network (CNN), and the success of graph-based representations was tested. This model, which is compared with CNN using 2D graph structure, has been observed to be more successful than other traditional methods. In order to make a transparent comparison between traditional methods and the generated graph representation learning approach, the connection matrices expressing the graphs were reduced to one dimension and the classification was made with the ANN method, and a sensitivity of %91.2 was obtained. The results show that the graph representation approach is more successful compared to traditional text embedding techniques. Among the traditional methods, it reveals that BERT outperforms other methods, FastText gives close results to BERT for attribute counts up to 500 words, while GloVe is far from competitive with the lowest classification performance. This thesis study stands out as the first study in the literature in which word embedding vectors are transformed into visibility graphs under variable attribute condition.

Key Word : Network Representation Learning, Text Classification, Deep Learning, Natural Language Processing.

Science Code : 92408

TEŞEKKÜR

Bu tez çalışmasının planlanmasında, araştırılmasında, yürütülmesinde ve oluşumunda ilgi ve desteğini esirgemeyen, engin bilgi ve tecrübelerinden yararlandığım, yönlendirme ve bilgilendirmeleriyle çalışmamı bilimsel temeller ışığında şekillendiren sayın hocam Doç. Dr. İlker TÜRKER' e sonsuz teşekkürlerimi sunarım.

Sevgili aileme manevi hiçbir yardımı esirgemeden yanımda oldukları için tüm kalbimle teşekkür ederim.

İÇİNDEKİLER

Sayfa

Sayfa

ŞEKİLLER DİZİNİ

Sayfa

ÇİZELGELER DİZİNİ

Sayfa

SİMGELER VE KISALTMALAR DİZİNİ

SİMGELER

x	: giriş değeri
y	: çıkış değeri
W	: ağırlık
h	: aktivasyon fonksiyonu
$\sum$	: toplam fonksiyonu
O	: gözlenen frekans
E	: beklenen frekans
u	: satır
v	: sütun
P	: nominal ağırlık fonksiyonu
n	: terim sayısı
N	: doküman sayısı

KISALTMALAR

TF-IDF	: Term Frequency – Inverse Document Frequency (Terim Sıklığı -Ters Belge Frekansı)
SVM	: Support Vector Machine (Destek Vektör Makinesi)
BERT	: Bidirectional Encoder Representations from Transformers (İki Yönlü Kodlayıcı Temsilleri Transformer)
LSTM	: Long Short – Term Memory (Uzun Kısa Süreli Bellek)
Bi-LSTM	: Bidirectional Long Short Term Memory (İki Yönlü Uzun Kısa Süreli Bellek)
CNN	: Convolutional Neural Network (Evrişimli Sinir Ağı)
DP	: Doğru Pozitif
DN	: Doğru Negatif
YN	: Yanlış Negatif
YP	: Yanlış Pozitif
NLP	: Natural Language Processing (Doğal Dil İşleme)
KNN	: K-Nearest Neigbors (K-En Yakın Komşu)
LR	: Lojistik Regresyon
YSA	: Yapay Sinir Ağları
AI	: Artificial Intelligent (Yapay zekâ)
CBOW	: Continuous Bag of Words (Sürekli Atlama-Gram)
GloVe	: Global Vectors for Word Representation (Kelime Temsili için Küresel Vektörler)
MLP	: Multi-Layer Perceptron (Çok Katmanlı Algılayıcı)
ReLU	: Rectified Linear Unit (Doğrusal Düzeltme Birimi)
GCN	: Graph Convolutional Network (Grafik Evrişimli Ağ)

BÖLÜM 1

GİRİŞ

Doğal dil işleme, metinleri belirli kategorilere ayırmak için etkili araçlar sunar [1]. Yapay Zekâ (AI) modelleri, Doğal Dil İşleme (NLP) prosesleri içerisinde sınıflandırma ve özellik çıkarma görevlerini birlikte ele alan güçlü yöntemler bütünüdür [2]. Metin konusu tanıma, e-posta sınıflandırması, yazar cinsiyet sınıflandırması, duygu analizi vb. gibi çeşitli uygulamalarda yaygın olarak kullanılmaktadır [3]. Metin işleme, dilin kendisinden kaynaklanan karmaşıklık nedeniyle zor bir görev olduğundan, her problem türü için çeşitli metin temsil yöntemleri önerilmiştir. Popüler yöntemler arasında (bunlarla sınırlı olmamakla birlikte) kelime/ifade frekansları, n-gram frekansları, kelime kümeleme, gizli anlam indeksleme ve işlevsel kelimeler yer alır [4]. Yine doğru öznitelik seçimleri de programın başarısı açısından oldukça önemlidir.

Melika Behjati çalışmasında; sinir ağlarının metin sınıflandırmasında yüksek başarılar elde etmesine rağmen veri setine eklenen küçük bir zıt kelimede bile doğruluk oranlarının yüzde 90'lardan yüzde 50'lere düştüğünü gözlemlemiştir. AG News veri seti giriş dizinlerinin başına rakip kelimeler eklemiş ve kelime sayıları arttıkça doğruluk oranlarının düştüğü sonucunu bulmuştur. Doğruluk oranı 1 zıt kelime eklendiğinde 93.42'den 49.72'ye düşmüştür [5]. Metin biçimli verilere uygulanan temsil yöntemleri, metin işlemenin kritik noktalarından biridir. Kelimelerin cümle içindeki kullanımları üzerinden sayısal vektörlere dönüştürülmesine en basit tanımıyla kelime gömme denir [6]. Kelime gömmeleri denetimsiz bir şekilde üretilir ve her kelimenin anlamı ile ilgili sabit boyutlu bir vektör oluşturmak için büyük bir korpus ile bir yerleştirme modeli eğitilir. Bunun sonucu olarak benzer kelimelerin gömme vektörleri de özellik uzayında benzer şekilde temsil edilir ve bu da herhangi bir dildeki metinlerin daha iyi anlaşılmasını sağlar [6,7]. Kelime gömme, farklı NLP sorunları için etkili bir yaklaşımdır. Kelime gömme ile ilgili literatürde pek çok çalışma

bulunmaktadır. Çalışmalar genellikle farklı veri setleri üzerinde kelime gömme yöntemlerini kendi aralarında veya makine öğrenmesi yöntemleri ile karşılaştırılması şeklinde karşımıza çıkmaktadır. Eddy ve arkadaşları 20 farklı haber konusuna sahip 19.997 dokümandan oluşan UCI KDD arşivini veri seti olarak kullanmışlardır ve Word2Vec, Glove ve FastText algoritmalarını karşılaştırmışlar ve sınıflandırma algoritması olarak CNN derin öğrenme yöntemi kullanılmıştır. Sonuçlar incelendiğinde doğruluk kriteri açısından %97.2 oranla FastText çok daha iyi bir performans sergilemiştir. Ardından Glove %95.8 ve Word2Vec %92.5 ile takip etmektedir [8].

Başka bir çalışmada Bi-directional Encoder Representations from Transformers (BERT) algoritmasından faydalanarak metinlerdeki nefret söylemi tespiti hedeflenmiştir. Nefret söylemi çevrim içi ortamlarda artan bir sorun haline gelmektedir ve otomatik tespit yöntemleri bu gibi durumlarda oldukça önemli bir hale gelmektedir. BERT algoritması metin sınıflandırması için güçlü bir dil modelidir ve sonuçlar önerilen modelin yüksek bir tespit oranına sahip olduğunu göstermektedir. F-Skor üzerinden değerlendirilen çalışmada, Bert Large %96.46 ile en yüksek başarıma sahiptir. Veri seti olarak Google News, Wikipedia ve Twitter verilerinden yararlanılmıştır. F-Skor üzerinden değerlendirilen karşılaştırmada BERT algoritması geleneksel yöntemlerden çok daha iyi bir performans sergilemiştir [9] .

Diğer bir çalışmada Arapça kelimelerden duygu analizi çalışılmış ve çalışmada 5 farklı veri seti kullanılmıştır. Baseline, FastText, Word2Vec, Glove ve BERT kelime vektörlerinin doğruluk kriteri açısından performansları incelendiğinde her bir ayrı veri seti için BERT algoritmasının daha başarılı olduğu incelenmiştir. Çalışmada CNN sınıflandırma yöntemi kullanılmıştır [10]. Covid-19 tweetlerinden intihar tahmini yapmayı amaçlayan başka bir çalışma, kelime yöntemlerinden Word2Vec, FastText, Glove yöntemleri üzerinden ayrı ayrı Long-Short Term Memory (LSTM), CNN ve Bidirectional LSTM (Bi-LSTM) derin öğrenme yöntemlerini karşılaştırmayı hedeflemiş olup F-Skor açısından baktığımızda her ne kadar açık ara bir fark olmasa da Bi-LSTM yöntemi ile daha başarılı oldukları gözlemlenmektedir [11].

Aydoğan ve arkadaşları tarafından, Türkiye Büyük Millet Meclisi tutanakları ve Wikipedia Türkçe makalelerinden oluşturulan yaklaşık 60 GB büyüklüğünde bir veri seti kullanarak geleneksel kelime temsil yöntemleri karşılaştırılmıştır. Çalışma kapsamında Word2Vec yönteminin iki modeli olan Sürekli Kelime Torbası (Continuous Bag of Words -CBOW) ve Sürekli Atlama-Gram (Continuous Skip-Gram) algoritmaları da analiz edilmiş olup çalışmada kullanılan veri seti oldukça büyük olduğu için CBOW algoritması daha iyi sonuç verirken; derlemin daha küçük bir kısmıyla çalışmalar yapıldığında Skip-Gram algoritmasının daha başarılı olduğu gözlemlenilmiştir [1].

Kullanılacak olan veri setine en uygun kelime gömme yöntemini bulmanın yanında özellik seçimi de metin sınıflandırması için oldukça önemlidir. Özellik seçimi, kelime gömme vektörleri arasında en fazla bilgi taşıyan ve sınıflandırmada etkili olan özellikleri seçerek modelin daha iyi performans göstermesini sağlamaktadır. Böylece gereksiz gürültü ve hesaplama maliyeti de giderilmiş olur. Setianga ve arkadaşları, metin sınıflandırması çalışmalarında N-Gram ve ki-kare özellik seçimi yöntemlerini Destek Vektör Makinesi (SVM) makine öğrenmesi yöntemiyle birlikte kullanarak daha yüksek bir doğruluk oranı sunmuşlardır. Veri setinde kullanılan özellik sayısı azaldığı için modelin daha hızlı çalışması da sağlanmıştır [12]. Başka bir çalışmada veri setindeki metin ön işlemeye tabi tutulduktan sonra seçilen özelliklerin sayısı sırası ile 100, 400, 700, 1000 ve 2000 olarak seçilmekte olup özellik ağırlıklandırması için Term Frequency – Inverse Document Frequency (TF-IDF) yönteminden yararlanılmaktadır. Çalışma, ki-kare ve bilgi kazancı yöntemlerini aynı anda karşılaştırmaktadır. Her iki yöntemde de özellik arttıkça doğruluk oranının arttığı gözlemlenmektedir. Ki-kare yönteminde 100 özellik için %85.69 doğruluk oranına ulaşılırken 1300 özellik için %96.74 doğruluk oranı ile ideal doyuma ulaşmıştır. 2000 özellik sayısında bu oran %95.03'e gerilemiştir. Çalışmada sınıflandırma modeli olarak SVM kullanılmıştır [13]. Başka bir çalışmada da özellik uzayının yüksek boyutluluğundan kurtulmak için ki-kare testi önerilmiştir. Yaklaşık 1100 belgeden oluşan ACM Digital Librray veri setini kullanan çalışma %92 F-Skor oranı ile metin sınıflandırma yöntemlerinin performansını iyileştirdiğini, sayıyı ve özellikleri azaltırken; sınıflandırıcıların yüksek performansının korunduğunu gözlemlemiştir [14]. Bu tez çalışmasında da ki-kare özellik seçimi yönteminden yararlanılmaktadır.

Sınıflandırma da en etkili 1500 kelime tespit edilmiş olup, bu kelimeler sırasıyla 100, 200, 300, 400, 500, 600, 700, 800, 900, 100, 1100, 1200, 1300, 1400, 1500 adet şeklinde modele gönderilip doğruluk ve F-Skor açısından karşılaştırılmıştır.

Son birkaç yıldaki çalışmalara baktığımızda literatürde metin sınıflandırması için AG News veri setini kullanan pek çok yayın mevcuttur. Zhang [15] karakter düzeyinde evrişimli ağlar kullanarak geleneksel yöntemlerle kıyaslama yapmıştır. Bu kıyaslamada büyük boyutlu veri setleri tercih edilmiştir. Tercih edilen veri setlerinden biri olan AG News veri seti kullanılan modeller arasında en iyi performansı "N-Gram TF-IDF" modelinde 7.64 hata oranı ile göstermiştir. Başka bir çalışmada AG News veri seti kullanılarak "FastText" isminde hızlı metin sınıflandırıcı tasarlamışlardır. Yapılan çalışma derin öğrenme yöntemleriyle karşılaştırıldığında "doğruluk" kriteri bakımından başarımı aynı seviyede fakat daha hızlı olduğu görülmüştür. Bir dakikadan daha az bir sürede yarım milyon cümleyi sınıflandırabilmektedir. AG News veri setinin doğruluk kriteri açısından başarısı %91.5' tir [16].

Wang ve arkadaşları, çalışmalarında klasik yöntemde NLP tekniklerinin cümleler için işe yarasa da, kısa metinler için kolay uygulanamadığından bahsetmektedir. Kullanılan AG News külliyatında kısa metinleri test etmek için makaleler kaldırılıp sadece başlıkları kullanılmıştır.

AG News veri seti kullanılan modeller arasında en iyi performansı "KPCNN" modeli göstermiş olup başarı oranı %88.36'dır [2]. Başka bir çalışmada metinlerin yerel özelliklerini çıkarmak için CNN ve çıkarılan özellikleri bağlamak için LSTM derin öğrenme yöntemleri kullanılmıştır. Makalede CNN ve LSTM yöntemlerini birleştirdikten sonra AG News veri seti üzerinde sınıflandırma başarısını denemişler ve %93.38 doğruluk oranı elde etmişlerdir. Ayrıca 2 adet Çince veri seti ve 5 adet İngilizce veri seti üzerinde bu modeller diğer çalışılan yöntemlerle karşılaştırılmıştır [17]. Sachan, çalışmasında maksimum olasılık eğitimi kullanan "bi-LSTM" modelini kullanmış olup çapraz entropi kaybı ile bu model eğitildiğinde sınıflandırma başarısı oldukça yüksek çıktığı görülmüştür. Özellikle AG News veri seti %5.62 hata oranıyla bu yöntem için en iyi sonucu veren veri seti olmuştur [3]. Izmailov, yarı denetimli bir olasılıksal sınıflandırma yöntemi olan FlowGMM'yi önermiştir ve AG News veri seti

üzerinde K-En Yakın Komşu (KNN) (%51.3), Lojistik Regresyon (LR) (%78.9) gibi klasik modellere kıyasla %84.8 doğruluk oranı ile en iyi sonucu elde etmiştir [3]. Amasyalı ve arkadaşları [18] çalışmalarında Türkçe metin sınıflandırma için 6 Türkçe veri seti üzerinde temsil yöntemlerinin performanslarını karşılaştırmışlardır. Çalışmalarında uyguladıkları kelime temsil metotları arasında en başarılı performansı N-Gram yöntemi göstermiştir.

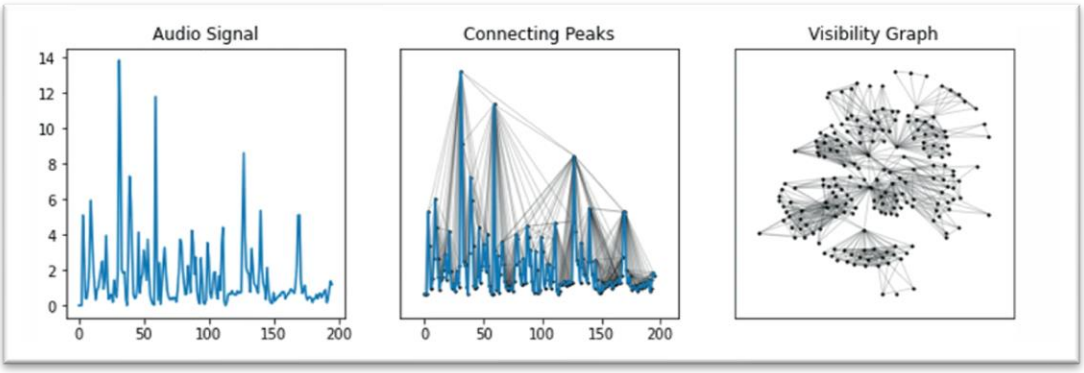
Klasik kelime gömme yöntemlerinin yanında graf tabanlı metin sınıflandırma çalışmaları da çok sık olmasa da karşımıza çıkmaktadır. Graf tabanlı metin sınıflandırma, metinlerin anlam ilişkisini yakalayabilmek ve sınıflandırma başarısını arttırabilmek amacıyla kullanılır. Bu tip bir graf yapısında kelimeler düğümleri, kelimeler arasındaki ilişkiler ise graf kenarlarını temsil ederler [19]. Konu ile ilgili yapılan bir çalışma, graf tabanlı metin sınıflandırmanın geleneksel vektör tabanlı yaklaşımlardan daha iyi performans gösterebildiği ve metinlerin yapısını daha iyi temsil edebileceğini belirtmektedir [19].

Liang ve arkadaşları metin sınıflandırması için klasik yöntemin dışında bir Text GCN (Graph Convolutional Networks) adında graf yapısı önermektedirler. Makalede, bir metin veri seti için kelime birliktelikleri ve belge-kelime ilişkileri esas alınarak tek bir metin grafı oluşturulmaktadır. Yapılan deneysel çalışmalarda Text GCN'nin dış kaynaklı herhangi bir kelime gömme bilgisine ihtiyaç duymadan bile oldukça başarılı bir performans yakaladığı görülmektedir [20]. Model "20NG" veri seti üzerinde %86.34, "R8" veri seti üzerinde %97.07 başarıya sahipken "Ohsumed" veri seti üzerinde başarım oranı %68.36 olarak kalmıştır. Bu da modelin başarımının veri setine göre değişiklik gösterebileceği sonucunu vermektedir. Ayrıca model FastText yöntemi ile de karşılaştırılmış olup tanıtılan bu graf yöntemi daha başarılı bir sonuç yakalamıştır.

Başka bir çalışmada "deep graph kernels" kullanılarak, yapısal verilerin karşılaştırılması ve analizi için yeni bir yöntem sunulmaktadır. Graf verilerinin benzerliklerini hesaplamak için kullanılan yaygın bir yöntem olsa da derin öğrenme tekniklerinin graf verileri üzerindeki etkisi ve kullanımı çok daha sınırlıdır. Çalışma derin öğrenme tekniklerinin graf çekirdeklerine entegrasyonu ile çok daha güçlü ve

esnek bir karşılaştırma yöntemi sunmaktadır. Çalışma farklı biyoinformatik ve sosyal ağ veri setleri üzerinde denenmiştir [21]. Benzer bir yöntemi Perozzi ve arkadaşları da önermiştir. Makalede önerilen "DeepWalk" yöntemi bir sosyal ağ grafı olarak temsil eder. Rasgele gezinmelerle oluşturulan yürüyüşler, kelime gömme vektörü olan Skip-Gram ile işlenir ve bu şekilde her kullanıcıya kelime benzeri bir temsil atanır [22].

Kelime temsil yöntemlerinin metin özelinde kullanımında görünürlük graflarından faydalanıldığı görülmektedir. Görünürlük grafları, verilen bir zaman serisi veri seti üzerinde yapılan grafiksel bir dönüşümdür. Şekil 1.1' de bir ses sinyaline ait veri setinin görünürlük grafiği gösterilmektedir [23].



Şekil 1. 1. Ses sinyallerinin görünürlük grafiklerine dönüşümü.

Bu tez çalışmasında literatürdeki önemli sayıdaki çalışma gibi veri kaynağı olarak AG News veri seti kullanılmış, 4 sınıflı metin sınıflandırma problemi için optimum çözümler araştırılmıştır. Ki-kare yöntemi ile sınıflar arasındaki kullanım sapması en yüksek kelimeler 100 ile 1500 arasında değişen sayıda seçilerek kullanılmıştır. Bu kapsamda istatiksel bir yöntem olan ki-kare testi, metin sınıflandırmada öne çıkan öznitelikleri, sınıf bazındaki varyasyonlarına dayalı olarak seçmeyi hedeflemektedir. Böylece çalışmada öznitelik uzayını daraltarak model karmaşıklığını azaltıp, performansı arttırmak amaçlanmıştır. Word2Vec (Skip-Gram, CBOW), FastText, GloVe, BERT kelime gömme teknikleri için ayrı ayrı gerçekleştirilen deneylerde kelime temsillerinin birleştirilmesi ile elde edilen cümle ve doküman temsilleri standart bir yapay sinir ağına girdi olarak verilmiştir. Bu sayede farklı kelime gömme

tekniklerinin farklı öznitelik sayısı koşulu altındaki başarıları irdelenmiştir. Elde edilen doküman temsillerine görünürlük grafı yaklaşımı uygulanarak oluşturulan 2 boyutlu graf temsilleri CNN modeli ile sınıflandırılmıştır. Aynı zamanda, grafları ifade eden bağlantı matrislerine ait düğüm derece listeleri hesaplanarak YSA ile sınıflandırılmıştır. Bu iki yöntemle de saf cümle temsilleri ile elde edilen başarının üzerinde bir sınıflandırma başarısı elde edilmiştir. Literatürü incelediğimizde; pek çok veri seti üzerinde klasik kelime temsil yöntemleri kullanılarak sınıflandırmalar yapıldığı, fakat kelime gömme vektörlerine uygulanan graf dönüşümleri ile sınıflandırma çalışması yapılmadığı gözlemlenmiştir. Çalışmamızda AG News veri seti üzerinde klasik kelime gömme yöntemleriyle birlikte bu kelime gömme vektörlerinin graflara dönüştürüldüğü bir çalışma da yapılarak başarımın arttırılması hedeflenmiştir.

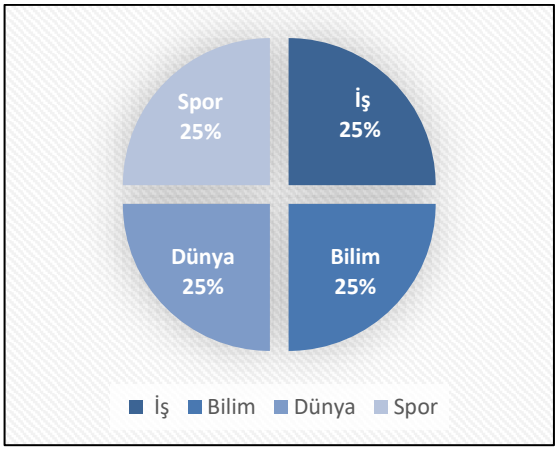
BÖLÜM 2

MATERYAL VE METOTLAR

2.1. KULLANILAN VERİ SETİ

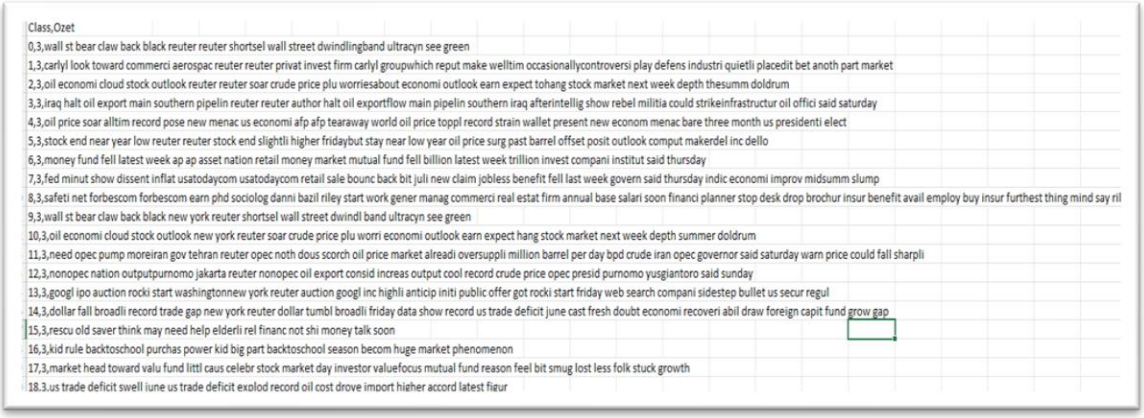
Bu tez çalışmasında ‘AG News’ adlı haber veri seti kullanılmıştır. AG News, bir milyondan fazla makaleden oluşan bir haber koleksiyonudur. Oluşturulan bu koleksiyon, Temmuz 2004'ten beri çalışan bir akademik haber arama motoru olan “ComeToMyHead” tarafından 2000'den fazla haber kaynağından toplanmıştır. AG News veri seti pek çok akademik topluluk tarafından sınıflandırma, kümeleme, bilgi erişimi, veri sıkıştırma, veri akışı amacıyla kullanılmaktadır [24] .

AG News veri seti, orijinal derlemden en büyük 4 sınıf seçilerek oluşturulmuştur. Her sınıf 30.000 eğitim örneği ve 1.900 test örneği içermektedir. Buna göre 120.000 doküman test verisi ve 7.600 doküman eğitim verisi olarak kullanılmaktadır [17]. Çalışmada kullanılan “train.csv” ve “test.csv” dosyaları, tüm eğitim örneklerini virgülle ayrılmış değerler olarak içerir. Veri setinde, dünya, spor, iş ve teknoloji haberleri olmak üzere 4 sınıf türü bulunmaktadır. İçlerinde sınıf indeksi, başlık ve açıklamaya karşılık gelen 3 sütun vardır [25]. Şekil 2.1’de veri setine ait sınıf dağılımları gösterilmektedir. Çalışmada sınıf isimlerine {1:'World News', 2:'Sports News', 3:'Business News', 4:'Science-Technology News'} olmak üzere 1, 2, 3, 4 olarak rakamlar atanmıştır.



Şekil 2. 1. AG News veri setine ait sınıf dağılımları.

Çalışmamızda, AG News veri seti işlenmemiş halinde bulunan "Title" ve "Description" kolonları birleştirilip "Ozet" adında yeni bir kolon oluşturulmuştur. Şekil 2.2' de ilk 20 satır için veriler gösterilmektedir. Böylece 3 sütun olan veriler, 2 sütuna indirgenip metin ön işleme adımlarına geçilmiştir.



Class,Ozet
0,3,wall st bear claw back black reuter reuter shortsel wall street dwindlingband ultracyn see green
1,3,carlyl look toward commerci aerospac reuter reuter privat invest firm carlyl groupwhich reput make welltim occasionallycontroversi play defens industri quietli placedit bet anoth part market
2,3,oil economi cloud stock outlook reuter reuter soar crude price plu worriesabout economi outlook earn expect tohang stock market next week depth thesumm doldrum
3,3,iraq halt oil export main southern pipelin reuter reuter author halt oil exportflow main pipelin southern iraq afterintellig show rebel militia could strikeinfrastructur oil offici said saturday
4,3,oil price soar alltim record pose new menac us economi afp afp tearaway world oil price toppl record strain wallet present new econom menac bare three month us presidenti elect
5,3,stock end near year low reuter reuter stock end slightli higher fridaybut stay near low year oil price surg past barrel offset posit outlook comput makerdel inc dello
6,3,money fund fell latest week ap ap asset nation retail money market mutual fund fell billion latest week trillion invest compani institut said thursday
7,3,fed minut show dissent inflat usatodaycom usatodaycom retail sale bounc back bit juli new claim jobless benefit fell last week govern said thursday indic economi improv midsumm slump
8,3,safeti net forbescom forbescom earn phd sociolog danni bazil riley start work gener manag commerci real estat firm annual base salari soon financi planner stop desk drop brochur insur benefit avail employ buy insur furthest thing mind say ril
9,3,wall st bear claw back black new york reuter shortsel wall street dwindl band ultracyn see green
10,3,oil economi cloud stock outlook new york reuter soar crude price plu worri economi outlook earn expect hang stock market next week depth summer doldrum
11,3,need opec pump moreiran gov tehran reuter opec noth dous scorch oil price market alreadi oversuppli million barrel per day bpd crude iran opec governor said saturday warn price could fall sharpli
12,3,nonopec nation outputpurnomo jakarta reuter nonopec oil export consid increas output cool record crude price opec presid purnomo yusgiantoro said sunday
13,3,googl ipo auction rocki start washingtonnew york reuter auction googl inc highli anticip initi public offer got rocki start friday web search compani sidestep bullet us secur regul
14,3,dollar fall broadli record trade gap new york reuter dollar tumbl broadli friday data show record us trade deficit june cast fresh doubt economi recoveri abil draw foreign capit fund grow gap
15,3,rescu old saver think may need help elderli rel financ not shi money talk soon
16,3,kid rule backtoschool purchas power kid big part backtoschool season becom huge market phenomenon
17,3,market head toward valu fund littl caus celebr stock market day investor valuefocus mutual fund reason feel bit smug lost less folk stuck growth
18.3.us trade deficit swell iune us trade deficit exolod record oil cost drove imoort hiaher accord latest fiaur

Şekil 2. 2. AG News veri setine ait ilk 20 satır

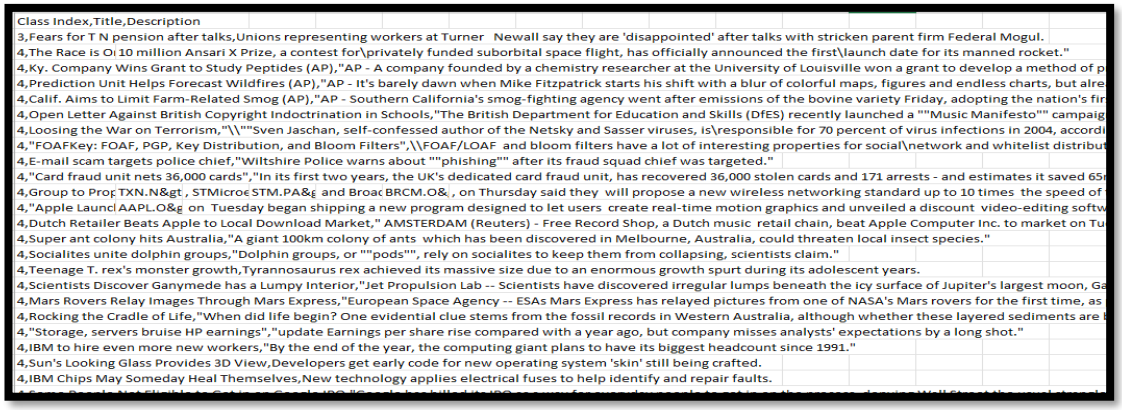
2.2. METİN ÖN İŞLEME ADIMLARI

Metin sınıflandırmasında en önemli aşamalardan birisi de metin ön işleme adımlarıdır. Ön işleme adımı ne kadar titiz ve iyi olursa sonuçların da aynı oranda güvenilir olması beklenir. Metin ön işleme şu adımlardan oluşur:

- ***Tokenization*:** Cümleleri kelimelere ayırma adımıdır.
- ***Lowercasing:*** Tüm kelimeleri küçük harfe çeviren işlem adımıdır. “Kalem” ile “kalem” aynı anlama gelir fakat küçük harflere çevirmediğimizde bu iki kelime vektör uzayı modelinde iki farklı kelime olarak temsil edilir. Bu da daha fazla boyut anlamına gelmektedir. Bu sebeple önemli bir adımdır.
- ***Stopwords Removal:*** Durdurma sözcükleri dokümanlarda en sık kullanılan edat, bağlaç gibi yapılardır. Bu kelimelerin tek başına bir anlamları yoktur ve bir önem ifade etmezler. Bu sebeple boyut azaltılması açısından dokümanlardan çıkartılması önemlidir.
- ***Stemming**:* Basit bir arama işlemi yaparken kelimenin ekli halinden çok kök halini önemseriz. Stemming kelimenin köklerine ayrıştırılması işlemdir.
- ***Lemmatization:*** Bu yöntemde bir çeşit kelimeleri köklerine ayırma yöntemidir fakat lemmatization, kelimelerin morfolojik analizini dikkate alır. Lemmatizasyon, tokenizasyon gibi, metin ön işlemede temel bir adımdır. Mevcut çeşitli diller göz önüne alındığında, lemmatization işlemi her dilde farklılık gösterebilir. [26].

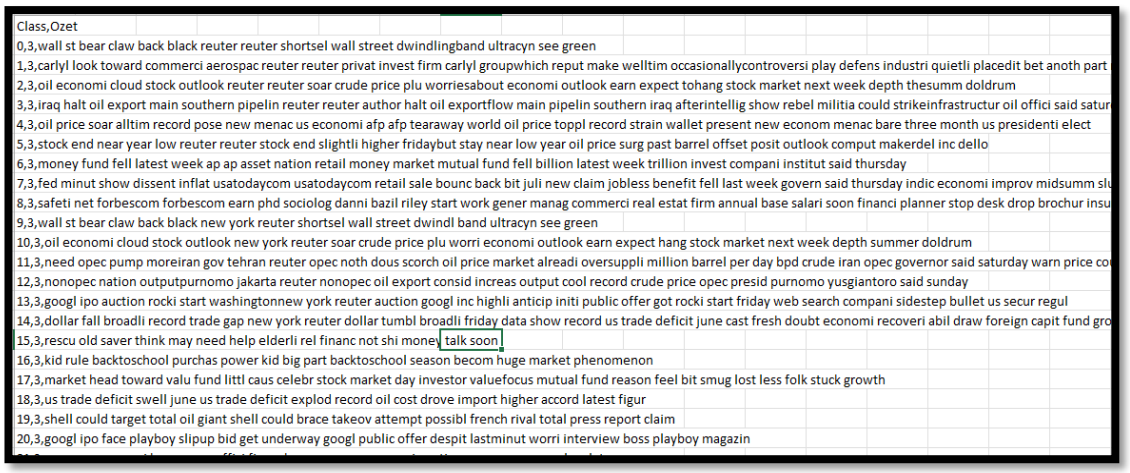
Bahsedilen metin ön işleme adımları boyut indirgemek için de yaygın olarak kullanılmaktadır. Bu adımlara ek olarak noktalama işaretleri, url adresleri, özel karakterler ve sayıların silinmesi de metin ön işleme adımları arasındadır. Bu adımlar Python programlama diline ait “Regex”, “NLTK” veya “Gensim” kütüphaneleriyle rahatlıkla gerçekleştirilebilmektedir.

Çalışmamızda bu adımları kullanarak ham verilerden işlenmiş veriler elde edilmektedir. Şekil 2.3’te de çalışmada kullandığımız AG News külliyatına ait ham veriler, Şekil 2.4’te ise çeşitli metin ön işleme adımlarından geçerek temizlenmiş olan veriler gösterilmektedir.



Class Index,Title,Description
3,Fears for T N pension after talks,Unions representing workers at Turner Newall say they are 'disappointed' after talks with stricken parent firm Federal Mogul.
4,The Race is Oi 10 million Ansari X Prize, a contest for\privately funded suborbital space flight, has officially announced the first\launch date for its manned rocket."
4,Ky. Company Wins Grant to Study Peptides (AP),"AP - A company founded by a chemistry researcher at the University of Louisville won a grant to develop a method of p
4,Prediction Unit Helps Forecast Wildfires (AP),"AP - It's barely dawn when Mike Fitzpatrick starts his shift with a blur of colorful maps, figures and endless charts, but alre
4,Calif. Aims to Limit Farm-Related Smog (AP),"AP - Southern California's smog-fighting agency went after emissions of the bovine variety Friday, adopting the nation's fir
4,Open Letter Against British Copyright Indoctrination in Schools,"The British Department for Education and Skills (DfES) recently launched a ""Music Manifesto"" campaig
4,Loosing the War on Terrorism,"\\""Sven Jaschan, self-confessed author of the Netsky and Sasser viruses, is\responsible for 70 percent of virus infections in 2004, accordi
4,"FOAFKey: FOAF, PGP, Key Distribution, and Bloom Filters",\\FOAF/LOAF and bloom filters have a lot of interesting properties for social\network and whitelist distribut
4,E-mail scam targets police chief,"Wiltshire Police warns about ""phishing"" after its fraud squad chief was targeted."
4,"Card fraud unit nets 36,000 cards","In its first two years, the UK's dedicated card fraud unit, has recovered 36,000 stolen cards and 171 arrests - and estimates it saved 65
4,Group to Proę TXN.N&gt , STMicroє STM.PA&ę and Broaє BRCM.O&¸ , on Thursday said they will propose a new wireless networking standard up to 10 times the speed of
4,"Apple Launcl AAPL.O&ę on Tuesday began shipping a new program designed to let users create real-time motion graphics and unveiled a discount video-editing softw
4,Dutch Retailer Beats Apple to Local Download Market," AMSTERDAM (Reuters) - Free Record Shop, a Dutch music retail chain, beat Apple Computer Inc. to market on Tu
4,Super ant colony hits Australia,"A giant 100km colony of ants which has been discovered in Melbourne, Australia, could threaten local insect species."
4,Socialites unite dolphin groups,"Dolphin groups, or ""pods"", rely on socialites to keep them from collapsing, scientists claim."
4,Teenage T. rex's monster growth,Tyrannosaurus rex achieved its massive size due to an enormous growth spurt during its adolescent years.
4,Scientists Discover Ganymede has a Lumpy Interior,"Jet Propulsion Lab -- Scientists have discovered irregular lumps beneath the icy surface of Jupiter's largest moon, Ga
4,Mars Rovers Relay Images Through Mars Express,"European Space Agency -- ESAs Mars Express has relayed pictures from one of NASA's Mars rovers for the first time, as
4,Rocking the Cradle of Life,"When did life begin? One evidential clue stems from the fossil records in Western Australia, although whether these layered sediments are
4,"Storage, servers bruise HP earnings","update Earnings per share rise compared with a year ago, but company misses analysts' expectations by a long shot."
4,IBM to hire even more new workers,"By the end of the year, the computing giant plans to have its biggest headcount since 1991."
4,Sun's Looking Glass Provides 3D View,Developers get early code for new operating system 'skin' still being crafted.
4,IBM Chips May Someday Heal Themselves,New technology applies electrical fuses to help identify and repair faults.

Şekil 2.3. AG News veri setine ait işlenmemiş dokümanlar.



Class,Ozet
0,3,wall st bear claw back black reuter reuter shortsel wall street dwindlingband ultracyn see green
1,3,carlyl look toward commerci aerospac reuter reuter privat invest firm carlyl groupwhich reput make welltim occasionallycontroversi play defens industri quietli placedit bet anoth part
2,3,oil economi cloud stock outlook reuter reuter soar crude price plu worriesabout economi outlook earn expect tohang stock market next week depth thesumm doldrum
3,3,iraq halt oil export main southern pipelin reuter reuter author halt oil exportflow main pipelin southern iraq afterintellig show rebel militia could strikeinfrastructur oil offici said satur
4,3,oil price soar alltim record pose new menac us economi afp afp tearaway world oil price toppl record strain wallet present new econom menac bare three month us presidenti elect
5,3,stock end near year low reuter reuter stock end slightli higher fridaybut stay near low year oil price surg past barrel offset posit outlook comput makerdel inc dello
6,3,money fund fell latest week ap ap asset nation retail money market mutual fund fell billion latest week trillion invest compani institut said thursday
7,3,fed minut show dissent inflat usatodaycom usatodaycom retail sale bounc back bit juli new claim jobless benefit fell last week govern said thursday indic economi improv midsumm sl
8,3,safeti net forbescom forbescom earn phd sociolog danni bazil riley start work gener manag commerci real estat firm annual base salari soon financi planner stop desk drop brochur insu
9,3,wall st bear claw back black new york reuter shortsel wall street dwindl band ultracyn see green
10,3,oil economi cloud stock outlook new york reuter soar crude price plu worri economi outlook earn expect hang stock market next week depth summer doldrum
11,3,need opec pump moreiran gov tehran reuter opec noth dous scorch oil price market alreadi oversuppli million barrel per day bpd crude iran opec governor said saturday warn price co
12,3,nonopec nation outputpurnomo jakarta reuter nonopec oil export consid increas output cool record crude price opec presid purnomo yusgiantoro said sunday
13,3,googl ipo auction rocki start washingtonnew york reuter auction googl inc highli anticip initi public offer got rocki start friday web search compani sidestep bullet us secur regul
14,3,dollar fall broadli record trade gap new york reuter dollar tumbl broadli friday data show record us trade deficit june cast fresh doubt economi recoveri abil draw foreign capit fund gro
15,3,rescu old saver think may need help elderli rel financ not shi money talk soon
16,3,kid rule backtoschool purchas power kid big part backtoschool season becom huge market phenomenon
17,3,market head toward valu fund littl caus celebr stock market day investor valuefocus mutual fund reason feel bit smug lost less folk stuck growth
18,3,us trade deficit swell june us trade deficit explod record oil cost drove import higher accord latest figur
19,3,shell could target total oil giant shell could brace takeov attempt possibl french rival total press report claim
20,3,googl ipo face playboy slipup bid get underway googl public offer despit lastminut worri interview boss playboy magazin

Şekil 2.4. AG News veri setine ait metin ön işleme adımları sonrası işlenmiş dokümanlar.

2.3. KELİME TEMSİL YÖNTEMLERİ

Metin ön işleme adımları gerçekleştirildikten sonra, metnin sayısallaştırılması gerekmektedir. Bir doküman içerisindeki kelimelerin nasıl temsil edileceği en önemli noktalardan birisidir. En sade ifadeyle metinlerin anlamlarının da gözetildiği biçimde sayısal ifadelere dönüştürülmesine kelime temsili (word embedding) denilmektedir. Aynı dokümanın farklı yöntemlerle farklı sayısal değerlere dönüştürülmesi mümkündür [27]. Kelime temsil yöntemleri, frekans bazlı ve tahmin bazlı olmak üzere ikiye ayrılmaktadır.

2.3.1. Frekans Bazlı Kelime Temsil Yöntemleri

Frekans bazlı kelime temsil yöntemleri, metin verilerini temsil etmek ve analiz etmek için kullanılan bir tür NLP yaklaşımıdır. Bu yöntemler, metin belgelerini sayısal vektörler ile temsil etmeye dayanır. Genellikle makine öğrenimi, veri analizi, metin işleme gibi yöntemlerde kullanılır. En yaygın frekans bazlı kelime temsil yöntemleri, TF-IDF, One-Hot Encoding, Kelime N- Gramları yöntemleridir [28].

2.3.1.1. One -Hot Encoding

Dokümanlardan çıkarılan kelime setinin dokümanda olup olmaması durumuna göre matris oluşturulur. Tüm kelimelerin olduğu bir grafik yapısı oluşturulup doküman bazlı kontrol edildiğinde, eğer dokümanda ilgili kelime varsa 1 yoksa 0 atayarak oluşturulan bir matris yapısıdır. Örneğin D1, D2 ve D3, veri setine ait cümleler olsun. Çizelge 2.1'de örnek cümlelere göre tablo yapısı gösterilmektedir.

D1: Biz bugün okulda olacağız.
D2: Biz bugün sinemaya gideceğiz.
D3: Biz bugün hastanedeyiz.

Çizelge 2. 1 One-hot encoding

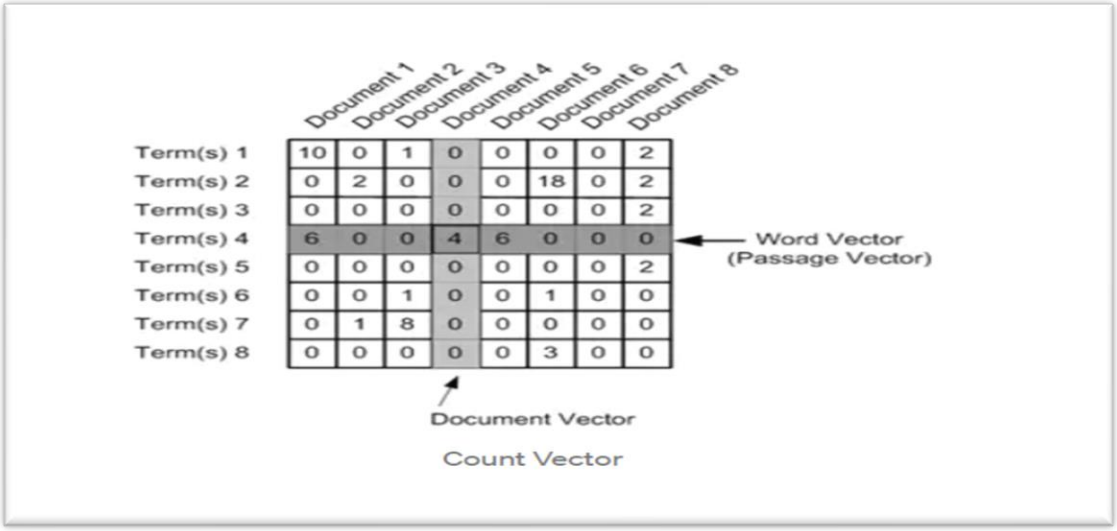
	Biz	Bugün	Okulda	Sinemaya	Hastanedeyiz	Gidiyorum
D1	1	1	1	0	0	1
D2	1	1	0	1	0	1
D3	1	1	0	0	1	0

2.3.1.2. Count Vector

Count Vector, NLP alanında kullanılan bir metin temsil yöntemidir. Bu yöntem de metinlerin sayısal temsillerini oluşturmak için kullanılmaktadır [28]. Dokümanlarda geçen kelime veya kelime setlerinin, dokümanlardaki geçme sıklığına göre oluşturulan vektörlerdir. Bu yöntemle metin belgelerindeki her bir kelimeye bir indeks atanır ve kelimenin dokümandaki tekrar sayısı çıkartılır. Bu şekilde metinler sayısal vektörlere

dönüştürülür ve makine öğrenimi algoritmalarında kullanılabilir yapıya dönüştürülürler. Her bir dokümana ait terim frekansları Şekil 2.5'te örnek olarak gösterilmektedir [29].

Count Vector yöntemi basit bir kelime temsil yöntemidir. Bu sebeple anlam bilgisi ve kelime sırasını tutma gibi özellikleri yoktur. Fakat kelime dağılımı ve frekans bazında önemli bilgiler verebilir bu nedenle birçok kelime tabanlı analizlerde kullanışlı olabilir [30].



Şekil 2. 5. Count vector mimarisi [29].

2.3.1.3. TF-IDF

TF-IDF (Term Frequency-Inverse Document Frequency) yöntemi, sık kullanılan kelime temsil yöntemlerinden birisidir. Metinlerin sayısal temsillerine dönüştürülmesi ve kelimenin önemini tespit etmek için kullanılır [31]. TF-IDF, bir kelimenin dokümandaki frekansını (TF) ve dokümandaki genel nadirliğini (IDF) birleştirerek o kelimenim önemini belirler. TF (Term Frequency), bir kelimenin dokümanda hangi sıklıkta geçtiğini belirtir. Örneğin bir belgede "gazete" kelimesi 5 defa geçiyorsa "gazete" kelimesinin TF değeri 5 olacaktır. IDF (Inverse Document Frequency), bir kelimenin dokümandaki nadirliğini belirtir. Dokümanda seyrek bulunan kelime yüksek IDF değerine sahipken, çok fazla kullanılan bir kelime düşük IDF değerine sahip olacaktır. Kelimenin sadece doküman içerisinde geçme oranından ziyade, diğer dokümanlar içerisinde de çok sık kullanılan bir kelime ise ağırlığı düşürülmektedir

[32]. Bu yöntemde logaritmadan da yararlanılmaktadır. Formülize etmek gerekirse TF değeri, terimin doküman içerisinde gözükme oranının dokümandaki terim sayısına bölümüdür. IDF ise Log(N/n) ile hesaplanmaktadır. Buradaki N, kaç doküman olduğu, n ise terimin kaç dokümanda görüldüğüdür. Yani TF değeri bağlaç gibi veya çok fazla tekrar eden kelimelerde bir artış sağlarken, IDF değerinden gelen 0 değeri, TF*IDF sonucunu 0 veya olabildiğince küçülterek değersiz bir hale döndürür ve sonuç olarak kelimenin önemini belirler.

Count Vector ile kıyaslandığında, Count Vector sadece bir kelimenin frekansını dikkate alırken, TF-IDF hem kelimenin frekansını hem de nadirliğini ele alır. Böylece seyrek bulunan ama önemli olan kelimelerin öne çıkmasını sağlar [33]. TF-IDF yöntemi, sınıflandırma, öneri sistemleri, bilgi keşfi gibi birçok doğal dil işleme uygulamalarında kullanılan etkili bir metin temsil yöntemidir [34].

2.3.1.4. Co-Occurrence Matrix

Co-Occurence Matrix yöntemi, doğal dil işleme alanındaki metin temsil yöntemlerinden birisidir. Bu yöntemdeki amaç, bir metin belgesindeki kelimelerin birbirleriyle olan ilişkilerini temsil etmektir. Bir başka deyişle kelimelerin birlikte gözükme miktarlarını tespit edebilmektedir [35]. “Co-occurrence” kelime ikilileri belirtilen doküman içerisindeki birlikte görülme sıklığı anlamına gelir [36]. Co-occurence matrix yöntemi, kelime bağlantılarına ait bilgileri verirken, kelimelerin sırasını ve anlam bilgisini dikkate almaz. Bu nedenle dokümanlardaki anlamsal bağları yakalayabilmek için diğer yöntemlerle birlikte kullanılabilir [37]. Örneğin veri setimizdeki dokümanlar aşağıdaki gibi verilsin.

Veri seti: Ben bugün okula gidiyorum / Ben bugün sinemaya gidiyorum / Ben bugün maça gidiyorum.

Veri setindeki bazı kelime ikilileri; “Ben Ben = 0, Ben Bugün = 3, Ben Okula = 1, Ben Sinemaya = 1” olacak şeklinde matris yapısında gösterilmiştir. Örneğe ait veriler Çizelge 2.2'de belirtilmektedir.

Çizelge 2.2.Occurrence matrix.

	Ben	Bugün	Okula	Sinemaya	Maça	Gidiyorum
Ben	0	3	2	2	1	2
Bugün	3	0	1	1	1	2
Okula	2	1	0	0	0	1
Sinemaya	2	1	0	0	0	1
Maça	1	1	0	0	0	1
Gidiyorum	2	2	1	1	1	0

2.3.2. Tahmin Bazlı Kelime Temsil Yöntemleri

Tahmin bazlı kelime temsil yöntemleri, metin verilerini işlemek ve temsil etmek için önceden eğitilmiş büyük dil modellerinden yararlanır. Bu tür modeller, kelime temsilini öğrenmek için büyük veri kümelerini kullanarak, dilin yapısını ve dil içindeki ilişkileri anlamayı öğrenirler [38]. En yaygın olarak kullanılan tahmin bazlı kelime temsil yöntemleri, Word2Vec, GloVe, FastText ve BERT yöntemleridir.

2.3.2.1. Word2Vec

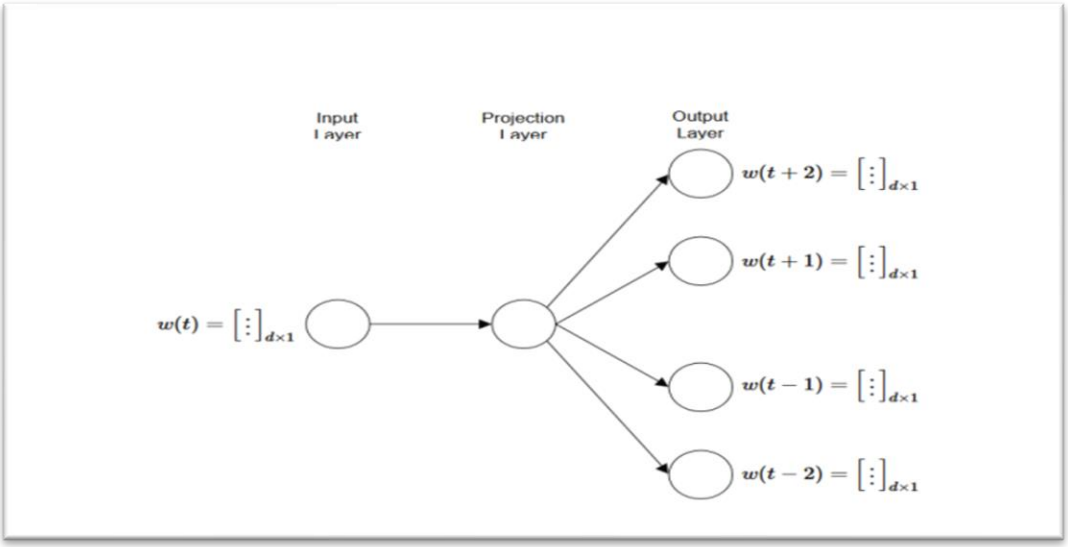
Tahmin bazlı kelime temsil yöntemlerinin ilki kelimeleri vektör uzayında temsil etmeye çalışan Word2Vec modelidir. Bu model doğal dil işlemede kullanılan kelime temsil yöntemlerindendir. Word2Vec yöntemi, kelimeleri yoğun vektörlerle temsil ederek kelime anlamlarını yakalamayı hedefler. Özellikle kelime benzerliklerini ve ilişkilerini yansıtmak için kullanılan oldukça popüler bir tekniktir [38].

Google araştırmacısı olan Tomas Mikolov ve ekibi tarafından 2013 yılında ortaya atılmıştır [39]. Word2Vec yönteminin 2 çeşit alt yöntemi vardır. Bunlar CBOW ve Skip-Gram modelleridir. Bu modeller büyük bir metin veri setini kullanarak kelime temsillerini öğrenirler. Bu iki yöntem yapısal olarak birbirine benzemektedir [40].

2.3.2.1.1. Skip-Gram

Skip-Gram modeli, Word2Vec yöntemi içinde yer alan kelime temsil modellerinden birisidir. Bu modeldeki amaç, hedef bir kelime verildiğinde, çevresindeki kelimeleri tahmin etmektir. Genellikle büyük bir metin havuzundaki kelime ilişkilerini ve birbirleriyle olan benzerliklerini yakalamak amacıyla kullanılır [38]. Skip-Gram modeli, ilgili kelimenin çevresine gelebilecek yakın kelimeleri tahmin etmeyi amaçlar. Örneğin, 'kedi' kelimesi için Skip-Gram modeli, "miyavlar" ve "koşar" gibi mantıklı olabilecek komşu kelimeleri tahmin etmeyi öğrenir.

CBOW ve Skip-Gram modelleri birbirlerinden çıktı ve girdi alma açısından farklılaşmaktadır [41]. CBOW modelinde "*window size"* parametresinin merkezinde olmayan kelimeler girdi olarak alınıp, merkezinde olan kelimeler çıktı olarak tahmin edilmeye çalışırken; Skip-Gram modelinde ise merkezdeki kelime girdi olarak alınıp merkezde olmayan kelimeler çıktı olarak tahmin edilmeye çalışılmaktadır. Bu işlem cümle bitene kadar devam etmektedir [42]. Bir cümleye uygulanan bu işlemler tüm cümlelere uygulanmakta ve böylece başlangıçta elimizde bulunan etiketlenmemiş veriye dönüştürme işlemi uygulanmış olmakta ve bu şekilde eğitilmeye hazır olmaktadır. Skip-Gram algoritması için giriş hedef kelimelerdir. Çıkışlar ise Şekil 2.6'da görüldüğü gibi hedef kelimeleri çevreleyen çıkışlardır.



Şekil 2. 6. Skip-Gram mimarisi [42].

Skip-Gram modeli, çoğunlukla kelime dağılımı modellerinin oluşturulması için kullanılan oldukça popüler bir algoritma modelidir. Bu nedenle pek çok kelime benzerliği, kelime analizi, kelime ilişkileri, duygu analizi ve öneri sistemleri gibi uygulamalar için tercih edilebilir [43].

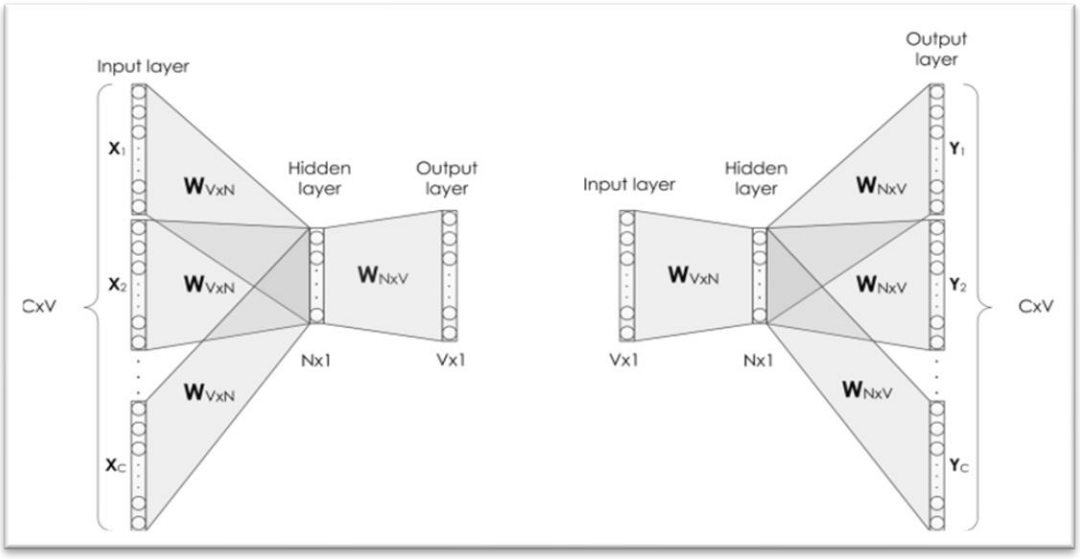
2.3.2.1.2. CBOW

CBOW modeli, Word2Vec yöntemi içerisinde yer alan kelime temsil yöntemlerinden biridir. Verilen bir kelimenin etrafındaki ilgili kelimeleri tahmin etmek amacıyla kullanılır [44]. Bu model, hedef kelimeye ait komşu kelimeleri dikkate alarak bir tahmin gerçekleştirir ve kelime temsili için, hedef kelimeye ait komşu kelimelerin ortalamasını kullanır [45]. Örneğin "kedi", "miyavlar", "koşar" gibi komşu kelimeleri içeren bir cümlede, "kedi" kelimesini tahmin etmek için "miyavlar" ve "koşar" kelimelerinin ortalaması hesaplanır.

CBOW modelleri genel olarak küçük veri setlerinde daha iyi çalışırken, büyük veri setlerinde Skip-Gram daha iyi çalışmaktadır [46]. Fakat bu durum genel bir kural değildir. Dilin yapısal özellikleri ve uygulamanın gereksinimleri gibi faktörler CBOW ve Skip-Gram modellerinin performansını etkileyebilir. Dolayısıyla en iyi sonuca ulaşabilmek için her iki modelin performansını karşılaştırmak önemlidir. CBOW, 2 veya daha çok anlamlı kelimeleri anlamakta iyi değilken; Skip-Gram 2 veya daha çok anlamlı kelimeleri daha iyi öğrenebilmektedir. İki yöntemin yapısal olarak karşılaştırılması Şekil 2.7'de gösterilmiştir.

CBOW modelinin katmanlarına baktığımızda giriş katmanı ile gizli katman arasında aktivasyon fonksiyonu yoktur. Yani ağırlıkların giriş katmanındaki düğümler ile olan çarpımına "relu", "tanh", "sigmoid" gibi lineer olmayan aktivasyon fonksiyonu uygulanmaz. Gizli katman ile çıkış katmanı arasında ise "softmax" aktivasyon fonksiyonu uygulanır. Böylece herhangi bir kelime giriş olarak alındığında, o kelimenin çıkışı **"e"** nin üzerine yazılarak tüm "**e^x"** lerin toplamına bölünür ve böylece bütün kelimelerin olasılık değerlerini içeren ve tekil kelime sayısı kadar bir büyüklüğe sahip bir vektör elde edilir. Bu sebeple daha kararlı bir temsil sağlar. Aynı zamanda daha az

hesaplama maliyetiyle çalışır. Bu özelliği CBOW' un küçük veri setlerinde daha iyi çalışmasını sağlar [38].



Şekil 2. 7. CBOW ve Skip-Gram mimarisi karşılaştırmalı gösterimi [46].

Sonuç olarak CBOW modeli genellikle küçük veri setleri ve hızlı eğitim süreleri için tercih edilirken, Skip-Gram modeli daha büyük ve daha ayrıntılı kelime temsilleri için tercih edilebilir [44].

2.3.2.2. Glove

Glove yöntemi, kelime dağılım tablolarından yararlanarak kelime temsillerini öğrenen bir algoritma yaklaşımıdır. Bu yöntem Pennington ve arkadaşları tarafından Stanford Üniversitesi'nde geliştirilmiştir [47]. Yönteme göre, bir kelimenin anlamı, kelimenin çevresindeki kelime dağılımıyla ilişkilendirilerek temsil edilir. Kelime dağılımı ifadesi, bir kelimenin bir metinde ne kadar sıklıkla birlikte kullanıldığını belirtir. Glove algoritması geniş bir metin havuzunda bahsedilen bu kelime dağılım tablolarını oluşturur. Sonrasında bu tablolar aracılığıyla kelime temsillerini öğrenir. Öğrenilen temsiller, kelime benzerliklerini ve ilişkilerini matematiksel olarak ifade etmek için vektör uzayında yer alır [48].

Glove yöntemi, Word2Vec yöntemlerine kıyasla birbirine yakın eş anlamlı sözcükleri bulmada daha başarılıdır. Bu yöntem kelime sayma konusunda eğitildiğinden istatistiksel hesaplamalarda da oldukça etkilidir. Denklem 2.1'de, Glove yöntemine göre optimize edilecek hata fonksiyonunu gösterir.

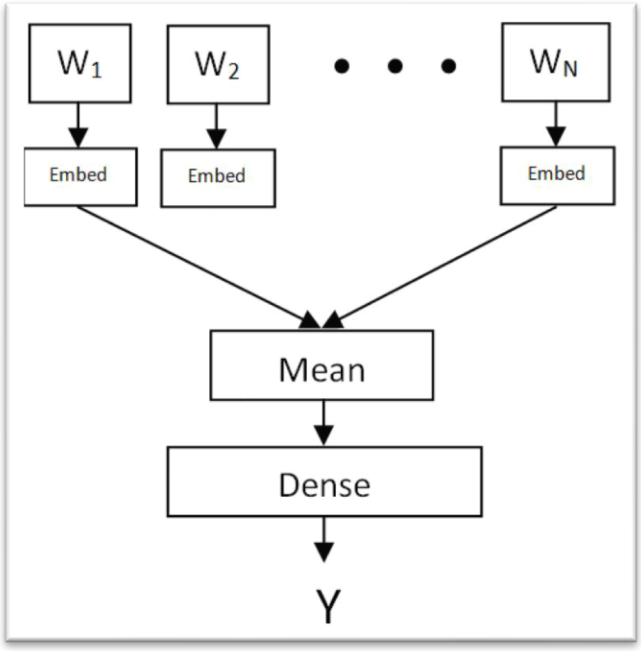
$$J = \frac{1}{2}\sum_{i,j=1}^{w} f(P_{i,j})(u_i^T v_j \log P_{i,j})^2 \quad (2.1)$$

u ve v sembolleri, global kelime sayıları ile oluşturulan matrisin sütun ve satır değerlerini temsil etmektedir. f(P_(i,j)) nominal ağırlık fonksiyonu iken, W oluşturulan sözlüğün boyutudur [47]. Doğal dil işlemede en çok kullanılan yöntemlerden birisidir. Bu temsiller kelime düzeyinde anlamsal ilişkileri yakalamak, metin sınıflandırma, kelime önerisi ve makine çevirisi gibi birçok alanda kullanılabilir [49].

2.3.2.3. FastText

Bu algoritma Facebook yapay zekâ araştırmacıları tarafından 2016 yılında geliştirilmiştir. Metin sınıflandırması için geliştirilen algoritma "Gensim" modelinde bir kütüphane şeklinde sunulmaktadır [50]. Aynı zamanda algoritma Word2Vec tabanlı bir modeldir. FastText yöntemi Skip-Gram yapısını kullanmaktadır. FastText algoritması, metinleri vektörlere dönüştürür. Word2Vec algoritmasından farkı bu yöntemin kelimelerin morfolojisine odaklanmasıdır. FastText algoritması kelimeleri N-gramlarına ayırmaktadır. N-gram söylemindeki n ifadesi, tekrar derecesini ifade eder. Yani n değerine göre kelimenin kaç karaktere bölüneceği belirlenir. Örneğin "kalem" kelimesi için tri-gram: "kal" ve "lem" harf dizimidir. Eğitimden sonra, tüm N-gramlar için kelime vektörleri üretilir. Sayı olarak daha az yaygın olan kelimeler, N-gram yöntemiyle daha etkili bir şekilde hesaplanabilir. Bazı veri tabanında olmayan kelimeler N-gramlar sayesinde temsil edilebilmektedir. N-gram sayısı kelime sayısından çok daha fazla olacağı için eğitim süresi uzamaktadır. Buna rağmen dokümanlarda az sıklıkta bulunan kelimeleri Word2Vec e kıyasla daha etkili bir şekilde temsil edebilmektedir [51]. FastText algoritması hızlıdır ve büyük verilerin

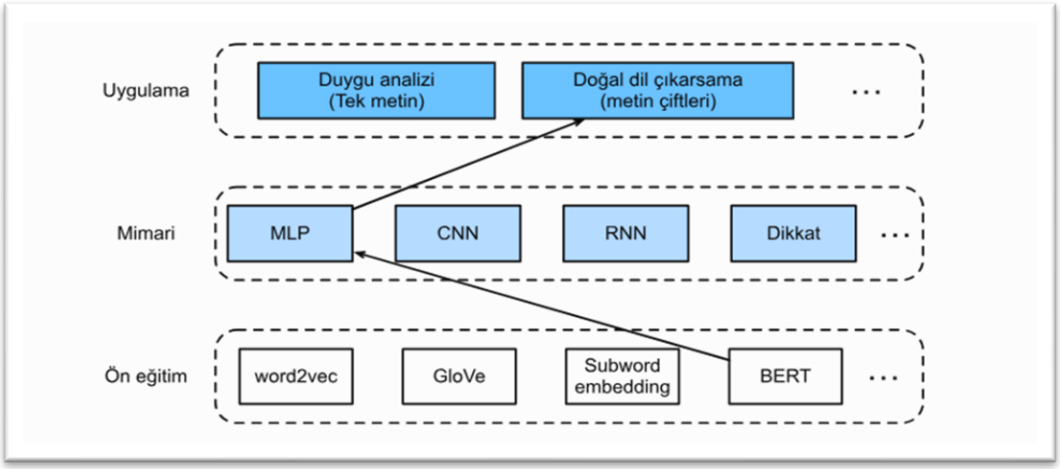
eğitilmesine izin vermektedir. Yüksek performans göstermesinin sebebi morfolojik yapısıdır. Şekil 2.8'de FastText modeli gösterilmektedir [52].



Şekil 2. 8. FastText mimarisi [52].

2.3.2.4. BERT

BERT algoritması, doğal dil işlemede kullanılan bir derin öğrenme modelidir. 2018 yılında Google tarafından açık kaynak olarak sunulmuştur [53]. BERT, önceden eğitimli bir model olduğu için büyük miktarda eğitim verisine ihtiyaç duymaktadır. Bu sebeple Wikipedia ve Google'ın Books Corpus veri seti üzerinde özel olarak eğitilmiştir. Ayrıca BERT algoritmasının, BERTBASE ve BERTLARGE olmak üzere iki versiyonu yayınlanmıştır [54]. BERT' nin başarısı, kelimeler arasındaki ilişkiyi bağlama göre öğrenen bir dikkat mekanizmasına sahip dönüştürücü bir derin öğrenme mimarisine dayanmasıdır.



Şekil 2. 9. BERT mimarisi [55].

Bu algoritma "transformer" adı verilen bir mimariyi kullanır. Her bir transfer bloğu, dikkat mekanizması ve Multi-Layer Perceptron (MLP) katmanlarından oluşur [56]. Transformer, dikkat mekanizmasını kullanarak bir kelimenin diğer bir kelime ile etkileşiminin modellenmesini sağlar. MLP katmanı ise, dikkat mekanizmasının çıktısını alır ve daha yüksek seviyeli özelliklerin çıkarılmasını sağlar. Bu katman, BERT modelindeki her bir transfer bloğunun çıkışında bulunur. MLP katmanı ayrıca iki adet tam bağlantılı katman ve Rectified Linear Unit (ReLU) aktivasyon fonksiyonu bulunan bir yapıya sahiptir. CBOW, Skip-Gram, FastText ve GloVe gibi geleneksel bağlamdan bağımsız kelime yerleştirme modellerinden farkı cümleyi hem soldan sağa hem de sağdan sola değerlendirmesidir Şekil 2.9'da BERT mimarisi gösterilmektedir [55].

2.4. ÖZNİTELİK SEÇİMİ

Metin madenciliği için temel problem, özellik uzayının yüksek boyutlu olmasıdır. Bir metin belgesi için özellik seti, tüm belgelerde yer alan bir dizi benzersiz terimdir. Özellik seçimi, öznitelik sayısını azaltan buna paralel olarak işlem hızını azaltan ve dolayısıyla daha yüksek verimlilik sağlayan bir yöntemdir.

Bir özellik seçim prosedürü uygulayarak, veriler nispeten daha küçük bir boyutta gösterilebilir. Bu arada, bellek kullanımı optimize edilir [10]. Gini İndeksi, Entropi veya Ki-Kare metrikleri, bir sınıflandırma görevi için kelimelerin önemini

değerlendirmek için yaygın olarak tercih edilir [57]. Bu çalışmada öznitelik seçim yöntemi olarak ki-kare ağırlık yöntemi kullanılmaktadır.

2.4.1 Ki-Kare Ağırlık Yöntemi

Ki-kare (x^2), istatiksel bir test yöntemi olup, değişkenler arasındaki ilişkiyi belirleyebilmek için kullanılır. Bir sınıfa ait kelimeler arasından hangi kelimelerin o sınıf için belirleyici olduğunun tespitini sağlar [10]. Literatürde, ki-kare testi olarak da geçmektedir. Özellikle kategorik verilerin analizinde yaygın olarak kullanılan güçlü bir yöntemdir. Rastgelelik ve büyük boyut özellikleri karşılandığında ham veriler üzerinde daha iyi performans gösterir [11]. Bu yöntem, beklenen sonuçlar ile gözlenen sonuçlar arasındaki tutarsızlıkları karşılaştırır. Test, bir hipotezin gözlenen verilere ne kadar uyduğunu gösterir. Denklemde gösterilen 'E' değeri beklenen değeri ve 'O' değeri gözlenen değeri temsil eder [10].

Spesifik olarak, belirli sınıflar boyunca bir kelimenin ortalama sıklığındaki sapmalar belirlenir. Sapması yüksek olan kelimeler, sınıf ayırt etme kapasiteleri daha yüksek olduğundan nitelik olarak seçilmiştir.

$$(\mathrm{X})^2 = \sum_{i=1}^{n} \frac{(O_i - E_i)^2}{E_i} \tag{2.2}$$

Ki-Kare istatistiği yukarıda belirtilen denklem 2.2 ile hesaplanır. Buna göre; x^2, ki-kare istatistiğini ifade eder. $\sum$, toplam sembolüdür ve tüm kategorilerin üzerindeki toplamı belirtir. O, gözlenen frekansı E ise beklenen frekansı ifade etmektedir. Bu şekilde, ki-kare testi, kategorik değişkenlerin bağımlılığını veya ilişkisini değerlendirmek için istatiksel bir araç olarak kullanılır [58].

Bu tez çalışmasında sınıflara ait etkili ve belirleyici kelimeleri bulabilmek adına ki-kare istatistiği kullanılmaktadır. Kelimelerin kullanım sıklığında sınıflar bazındaki sapmanın fazla olması, ilgili kelimenin sınıf-spesifik olarak etkili olduğunun bir göstergesi kabul edilmektedir. Dolayısıyla bu sapmayı ifade eden ki-kare ölçütü fazla

olan kelimelerin öznitelik olarak seçilmesinin sınıflandırma prosedürüne katkı sağlayacağı düşünülmektedir.

Şekil 2.10' da çalışmada kullanılan AG News veri setine ait kelimeler arasındaki en belirleyici 20 kelime gösterilmektedir. Şekildeki, O1, O2, O3 ve O4 ifadeleri ilgili kelimelerin o sınıfta kaç kez geçtiğini belirtmektedir.

	word	O1	O2	O3	O4	E	O1/300(	O2/300(	O3/300(	O4/300(	E	(O1-E)^2/E	(O2-E)^2/E	(O3-E)^2	(O4-E)^2	TOTAL_NORMALIZ
2	kill	5797	147	50	158	1538	0,19323	0,0049	0,00167	0,00527	0,05127	0,393131361	0,041935	0,04799	0,04127	**0,524**
3	iraq	5826	32	384	73	1578,75	0,1942	0,00107	0,0128	0,00243	0,05263	0,380873741	0,05051329	0,03014	0,04787	**0,509**
4	oil	833	23	6246	128	1807,5	0,02777	0,00077	0,2082	0,00427	0,06025	0,017513144	0,05872642	0,36331	0,05202	**0,492**
5	microsoft	25	0	750	5084	1464,75	0,00083	0	0,025	0,16947	0,04883	0,047172556	0,048825	0,01163	0,29809	**0,406**
6	price	469	78	5374	908	1707,25	0,01563	0,0026	0,17913	0,03027	0,05691	0,029936312	0,05182712	0,26251	0,01247	**0,357**
7	win	762	5158	396	412	1682	0,0254	0,17193	0,0132	0,01373	0,05607	0,016773682	0,23944859	0,03277	0,03196	**0,321**
8	stock	425	36	4269	319	1262,25	0,01417	0,0012	0,1423	0,01063	0,04208	0,018511588	0,03970922	0,23874	0,0235	**0,32**
9	profit	108	20	3710	397	1058,75	0,0036	0,00067	0,12367	0,01323	0,03529	0,028458892	0,03397093	0,2213	0,01379	**0,298**
10	game	201	5304	260	1927	1923	0,0067	0,1768	0,00867	0,06423	0,0641	0,051400312	0,19814805	0,04794	2,8E-07	**0,297**
11	minist	3574	35	386	77	1018	0,11913	0,00117	0,01287	0,00257	0,03393	0,213920629	0,03164011	0,01308	0,02899	**0,288**
12	compani	438	95	5423	4094	2512,5	0,0146	0,00317	0,18077	0,13647	0,08375	0,057095194	0,0775364	0,11238	0,03318	**0,28**
13	team	308	4043	155	612	1279,5	0,01027	0,13477	0,00517	0,0204	0,04265	0,024588049	0,19895616	0,03294	0,01161	**0,268**
14	softwar	24	0	765	3595	1096	0,0008	0	0,0255	0,11983	0,03653	0,034950852	0,03653333	0,00333	0,18993	**0,265**
15	iraqi	2771	5	62	13	712,75	0,09237	0,00017	0,00207	0,00043	0,02376	0,19812431	0,02342617	0,0198	0,0229	**0,264**
16	cup	70	2767	10	11	714,5	0,00233	0,09223	0,00033	0,00037	0,02382	0,019378598	0,19653633	0,02315	0,02309	**0,262**
17	coach	36	2692	16	4	687	0,0012	0,08973	0,00053	0,00013	0,0229	0,020562882	0,19505216	0,02185	0,02263	**0,26**
18	inc	69	11	4375	2315	1692,5	0,0023	0,00037	0,14583	0,07717	0,05642	0,051910433	0,05568572	0,14172	0,00763	**0,257**
19	palestinian	2589	4	4	4	650,25	0,0863	0,00013	0,00013	0,00013	0,02168	0,192682382	0,02140915	0,02141	0,02141	**0,257**
20	leagu	95	2712	16	29	713	0,00317	0,0904	0,00053	0,00097	0,02377	0,017855259	0,18681632	0,02271	0,02187	**0,249**
21	presid	4390	255	959	474	1519,5	0,14633	0,0085	0,03197	0,0158	0,05065	0,180756175	0,03507646	0,00689	0,02398	**0,247**

Şekil 2. 10. Ki-Kare hesaplama değerleri.

2.5. GRAF TEMSİLLİ ÖĞRENME

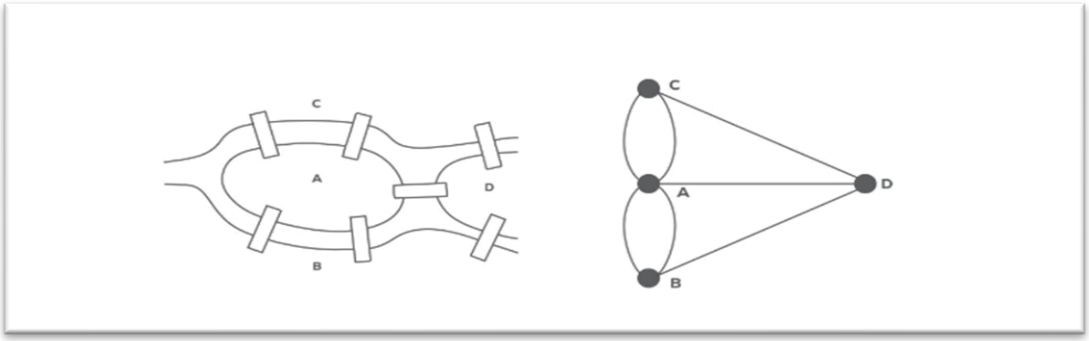
Graf temsilli öğrenme, genel olarak veri kümesinin grafiksel yapısını anlamak, görselleştirmek ve bu yapının çeşitli makine öğrenimi algoritmalarıyla analiz edilmesini ifade eder. Bu yaklaşım ağırlıklı olarak graf teorisi ve makine öğrenmesi olmak üzere iki ana unsura dayanır. Graf teorisi, düğümler arasındaki ilişkilerin bir araya gelmesiyle oluşan yapıları inceler. Makine öğrenmesi kısmında ise veri kümesinin graf yapısı kullanılarak düğümleri ve kenarları temsil eden matematiksel modeller geliştirilir. Bu modeller, düğümlerin özelliklerini ve kenarların ilişkilerini ifade eder. Daha sonra oluşturulan bu temsil yöntemi, sınıflandırma, kümeleme, tahmin gibi makine öğrenimi uygulamalarında kullanılabilir [21].

Graf temsilli öğrenme, sosyal ağ analizi, öneri sistemleri, biyoinformatik ve daha birçok alanda kullanılarak, veriler arasındaki gizli ilişkileri ve yapıları keşfetmeye yardımcı olur [59].

2.5.1. Graf Teorisi

Graf teorisi, nesnelerin veya kavramların birbirleriyle olan ilişkilerini modellemek ve analiz etmek amacıyla kullanılmaktadır [21]. Düğüm (vertex) adı verilen noktalardan ve bu noktaların arasındaki bağlantılardan (edge) oluşan ilişkisel veri topluluklarıdır. Bu teorem, İsviçreli Matematikçi Leonhard Euler'in 18.yy'da "Königsberg Köprüsü" problemi üzerinde yaptığı çalışmalar esnasında ortaya atılmıştır [60]. Euler şehrin krokisini çizmek yerine Şekil 2.11'deki bir graf modeli üzerinden problemi çözmeye çalışmıştır.

Graf teorisi, matematiksel araştırmalardan bilimsel uygulamalara kadar birçok alanda kullanılan disiplinlerarası bir çalışma haline gelmektedir. Özellikle ağ analizi, optimizasyon, veri tabanı sistemleri ve bilgisayar bilimleri (veri yapıları, yapay zekâ, görüntü işleme, ağlar) alanında sıklıkla kullanılmaktadır [60].



Şekil 2. 11. Şehir krokisi ve graf modeline uyarlanması.

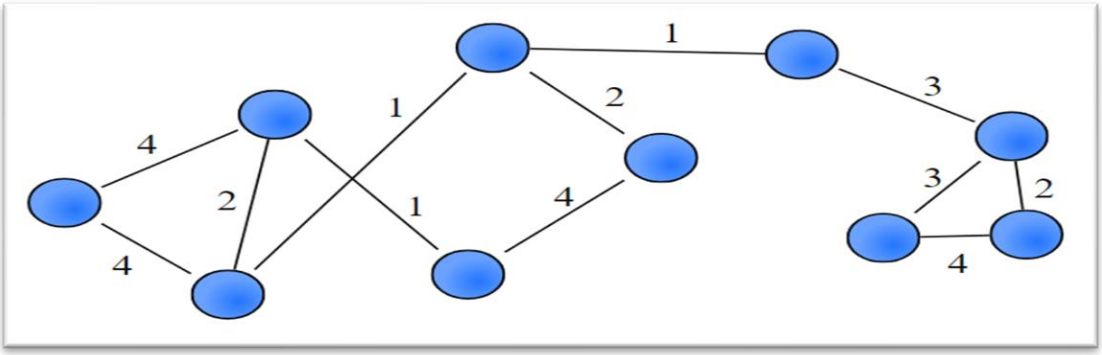
Grafların birçok çeşidi bulunmaktadır. Bunlardan bazıları aşağıdaki gibidir:

Yönsüz Graf: Bu graf çeşidinde düğümler arasındaki bağlantıların yönü bulunmamaktadır. Yönsüz graf modeline Facebook arkadaşlık ilişkilerini örnek olarak gösterebiliriz. Bu bağlantıda kural X kişisi Y kişisi ile arkadaş olduğunda, Y kişisi de X kişisi ile arkadaş olmuş olur [60].

Yönlü Graf: Bu graf çeşidinde düğümler arası yöne dayalı bir ilişki bulunmaktadır. X noktası Y noktası ile ilişkili olduğu halde Y noktasının X ile ilişkisi bulunmayabilir.

Bu duruma Twitter veya Instagramdaki takipleşme ilişkisi örnek olarak gösterilebilir [60].

Ağırlıklı Graf: Bu graf çeşidinde bağlantıların bir değeri bulunmaktadır. Bu bağlantılar, maliyet, zaman, uzunluk gibi özelliklere göre ağırlıklandırılır. Buna örnek olarak bir şehir haritası çizip, yol uzunluklarına göre bağlantıları ağırlıklandırabiliriz. Şekil 2.12' de bu graf modeli örnek olarak gösterilmektedir [60].



Şekil 2.12. Ağrlıklı graf örneği [60].

2.5.2. Karmaşık Ağlar

Karmaşık ağlar, graf teorisini temel alarak kullanan, gerçek dünyadaki yapıları ve karmaşık ilişkileri temsil eden, büyük miktarda düğüm ve bağlantı içeren ağ yapılarıdır. Genellikle düğümler ve bu düğümleri oluşturan bağlantılar olarak temsil edilmektedirler [61].

Karmaşık ağlar birçok disiplinlerarası alanda uygulanabilir. Örneğin sosyal ağ analizi, iletişim ağları, ulaşım ağları, biyolojik ağlar, kelime ağları, finansal ağlar ve bilgisayar ağları gibi çeşitli alanlarda karmaşık ağlar kullanılmaktadır. Bu ağlar gerçek dünyadaki karmaşık ilişkileri ve yapıları modellemek ve anlayabilmek için son derece önemlidirler [62]. Karmaşık ağların bazı temel özellikleri şu şekildedir:

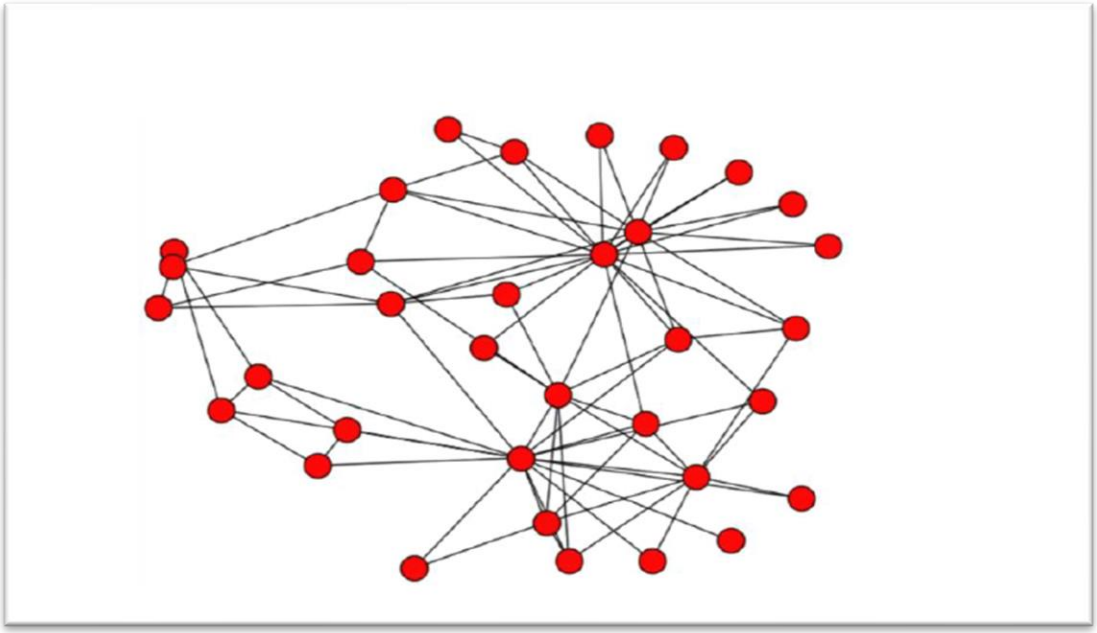
- ***Kümeleşme:*** Düğümlerin genel olarak belirli gruplar oluşturduğu gözlemlenmektedir. Bu gruplaşmalar benzer düğümlerin birbirleriyle daha sık bağlantı kurma eğiliminde olduğunu göstermektedir.

- ***Merkezilik:*** Karmaşık ağlarda bazı düğümler diğerlerinden daha merkezi veya önemli bir rol oynamaktadır. Bu merkezi düğümler ağdaki bilgi akışını daha fazla etkilemektedir.

- ***Kısa Yol Etkisi:*** Karmaşık ağlarda düğümler arası mesafelerin genellikle kısa olduğu görülmektedir. Bu bir düğümden diğerine ulaşmanın genellikle kısa bir yol ile mümkün olduğu anlamına gelmektedir.

- ***Ölçeklendirme:*** Karmaşık ağlarda düğümlerin bağlantı sayısının belirli bir dağılıma göre yapıldığı görülmektedir. Güç yasası olarak adlandırılan bu dağılım, bazı düğümlerin diğer düğümlere oranla çok daha fazla bağlantıya sahip olduğu gözlemlenmektedir. Fazla bağlantıya sahip az sayıda düğüm, az bağlantıya sahip çok sayıda düğüm ve düğümlerin bağlantı sayısına ait dağılımın güç yasasına uyumluluğu ile karakterize edilebilir [62].

Karmaşık ağlar model ağlar ve gerçel ağlar olmak üzere ikiye ayrılır. Model ağlar, düğümler arası bağlantıların önceden belirlendiği bağlantı prosedürleri kullanılarak yapay olarak oluşturulmasıdır [63]. Bu tür ağların bağlantı ve düğüm sayısı bağlantı protokolleri ile önceden belirlenmektedir. Gerçel ağlar ise gerçek dünyada var olan sistemlerin modellenmesi ile oluşturulan ağlardır. Gerçel ağlara, web sayfası ağı (World Wide Web), internet ağı, bilimsel atıf veya iş birliği ağları, biyolojik ağlar, telefon ağları, kelime ağları örnek gösterilebilir [64]. Şekil 2.13'te bir sosyal ağın en yaygın olarak kullanılan karmaşık ağ modeli ile temsili verilmiştir [61]. Bu örnekte, düğümler bir gruptaki bireyleri ve aralarındaki bağlantılar ise bu bireyler arasındaki etkileşimi temsil etmektedir.



Şekil 2.13. Bir gruptaki ilişkileri gösteren karmaşık ağ yapısı [61].

Kelime ağları, metinlerdeki kelime ilişkilerini modellemek ve analiz etmek için kullanılan ağ yapılarıdır [65]. Bu ağlar metinlerin birbirleriyle olan ilişkilerini göstermektedir. Kelime ağları doğal dil işleme, bilgi çıkarımı, metin sınıflandırma ve kelime önerme gibi analizlerde kullanılmaktadır. Bir metindeki kelime ağları kelimeler ve kelime ilişkilerinden oluşmaktadır [19]. Kelimeler düğümleri, kelime ilişkileri ise kenarları oluşturmaktadır. Kelime ağındaki düğümler metindeki farklı kelimeleri temsil eder. Düğümler genellikle kelime kökleri şeklinde ele alınırlar. Örneğin, "yürür", "yürüyor", "yürümek" gibi kelimeler aynı köke sahiptir. Bu sebeple tek bir düğümle temsil edilirler. Kelime ağındaki kenarlar ise metinler arasındaki ilişkiyi ifade eder. Örneğin iki kelimenin metinde birlikte geçmesi veya ardışık olması durumunda bu iki kelime arasında bir kenar oluşturulur [66].

2.5.3. Görünürlük Grafları (Visibility Graphs)

Hesaplamalı geometride ve robot hareket planlamasında bir görünürlük grafı, tipik olarak Öklid düzlemindeki bir dizi nokta ve engel için görülebilen konumların bir grafiğidir. Grafikteki her düğüm bir nokta konumunu temsil eder ve her kenar, aralarında görünür bir bağlantıyı temsil eder. Yani iki konumu birleştiren doğru parçası herhangi bir engelden geçmiyorsa grafikte aralarına bir kenar çizilir. Konum kümesi

bir çizgide yer aldığında, bu sıralı bir seri olarak anlaşılabilir. Görünürlük grafikleri bu nedenle zaman serisi analizi alanına genişletilmiştir [59].

Görünürlük grafları bir dizi noktanın bir düzlem üzerindeki konumlarını temsil eder. Bu noktalar genellikle engelleri ve sensörleri ifade etmektedir. Her nokta, diğer noktaları görüp göremediğini tespit edebilmek için birbirleriyle doğrusal bir bağlantıya sahip olup olmadığını kontrol eder. İki noktanın birbirlerini görebilmesi için aralarında hiçbir engel veya harici bir nokta olmamalıdır [23].

Görünürlük graflarından pek çok problemin çözümünde faydalanılabilir. Örneğin bir robot süpürgenin engelleri aşarak en iyi yolu bulabilmesi için bu tarz navigasyon problemlerinde kullanılabilir. Graf, robot süpürgenin bir noktadan diğerine geçerken engelleri en aza indirgemek veya tamamen önlemek için hareket etme yeteneğini sağlar. Görünürlük grafları ayrıca konum tabanlı hizmetlerde ve harita yönlendirme sistemlerinde en kısa yol veya en optimize yol hesaplanmasında kullanılabilir. Bir düzlem üzerindeki noktaların birbirini görebilme durumunu temsil eden bir graf yapısı sağlayan görünürlük grafları, bu özelliğiyle herhangi bir engeli aşmak, yol hesaplamaları veya görünürlük ile ilgili problemleri analiz etmek için kullanılır [67].

Görünürlük graflarının zaman serilerinde gösterdiği başarı, kelime uzayında da test edilebilir. Kelime vektörleri ve görünürlük graflarının birlikte kullanımı, metin sınıflandırma problemlerinde birbirini tamamlayıcı etki gösterme potansiyeline sahiptir. Örneğin, bir metin sınıflandırma probleminde metindeki kelimeleri, kelime vektörlerine dönüştürebilir ve bu vektörler, kolaylıkla görünürlük graflarına dönüştürülebilir. Oluşturulan graf yapısı, metin temsil vektörlerinin ifade ettiği bilgi içeriğini, vektörü oluşturan değerler arasında tanımladığı görünürlük örüntüleri ile zenginleştirerek, metindeki kelime ilişkilerini daha iyi anlamak için kullanılabilir. Bunun yanında, metindeki semantik olarak yakın kelimelerin birbirini gördüğü bir görünürlük grafı da oluşturulabilir [68].

Kelime vektörleri, kelime anlamını temsil etmekte ve metinleri sayısal bir şekilde kodlamaktadır. Görünürlük grafları ise metindeki kelimelerin birbirlerini görebilme durumunu temsil etmekte ve kelime ilişkilerini görselleştirmektedir. Böylece, kelime

vektörleriyle oluşturulan graf yapısı, metin analizinde kapsamlı ve derinlemesine bir anlam sağlamak için kullanılabilir. Şekil 2.17' de de çalışmada kullanılan görünürlük grafiğine bir örnek gösterilmektedir.

Bu grafiği elde edebilmek için veri setinde de bazı işlemler gerçekleştirilmektedir. Veri setimiz ilk olarak Şekil 2.14'te gösterildiği gibi ön işleme adımından geçmiş 96000 adet dokümana ait kelimeleri içermektedir. Şekil 2.15'teki gibi oluşturulan "x_train_vect" ve "x_tarin_test" adındaki eğitim ve test matrisleri için, eğer matris içinde veri varsa matrisin sütunlarının ortalaması hesaplanır ve bu ortalama sütun vektörünün her bir doküman için hesaplaması yapılır. Eğer matrisin içerisinde eleman yoksa uzunluğu 100 boyutlu olan sıfırlardan oluşan bir sütun vektörü oluşturulur ve "x_train_vect" listesine eklenir. Bu şekilde her bir matris için bir öznitelik vektörü elde edilmiş olur. Bahsedilen hesaplamaların ardından veri örneğimiz Şekil 2.16'daki gibi bir matris yapısına dönüştürülür.

```
X_train

91786     [grand, prix, snag, silverston, jaguar, may, g...
106728    [trade, loss, chines, firm, come, light, discl...
29318     [pakistan, offer, amnesti, terror, suspect, ap...
34895     [nation, lobbi, expand, secur, council, ap, ap...
63963     [check, point, post, rise, profit, ap, ap, int...
                                ...
17807     [report, bofa, job, cut, total, american, bank...
73951     [mubarak, arafat, discuss, isra, pullout, plan...
113478    [uk, polic, probe, alleg, chariti, fraud, plot...
89262     [china, strengthen, coal, mine, safeti, china,...
70389     [india, unawar, u, plan, nuclear, curb, new, d...
Name: stemmed, Length: 96000, dtype: object
```

Şekil 2. 14. Eğitim veri setine ait her bir doküman verisi.

```
x_train_vect
             [ 0.00292309, -0.05426001, -0.04707073, ..., -0.19808803,
              -0.06114947,  0.02507341],
             [ 0.36880082,  0.48395863, -0.5724662 , ..., -0.7713745 ,
              -0.10618289,  0.09079897]], dtype=float32)             ,
      array([[-0.7297594 ,  3.6865947 ,  1.0981237 , ...,  1.0076317 ,
               0.45539793, -0.6588892 ],
             [ 1.1023245 ,  0.40701404, -1.1604452 , ..., -0.6505025 ,
              -1.1682975 , -0.4932902 ],
             [ 0.26628333,  0.1916977 ,  0.2684202 , ...,  0.3452115 ,
               0.00756142, -0.30979478],
             ...,
             [ 0.93627834, -1.5273978 ,  0.07674674, ..., -0.15585795,
              -1.558551  , -0.560626  ],
             [ 0.19471045, -0.1866521 , -0.9868862 , ...,  0.74171156,
              -0.7232244 ,  0.5701646 ],
             [ 1.838169  ,  0.60972065,  1.7454134 , ...,  0.6038548 ,
               0.975205  , -0.46972132]], dtype=float32)             ,
      ...,
      array([[-0.4249855 , -0.15029238, -0.32818228, ..., -1.6387326 ,
               0.9390861 , -0.6230256 ],
             [-0.1637203 ,  1.3022591 ,  1.2651111 , ...,  1.2764771 ,
               1.2058313 , -1.6319885 ],
             [ 1.0210997 , -0.10506746,  0.3715519 , ...,  0.02391525,
               0.13178207, -2.0064764 ],
```

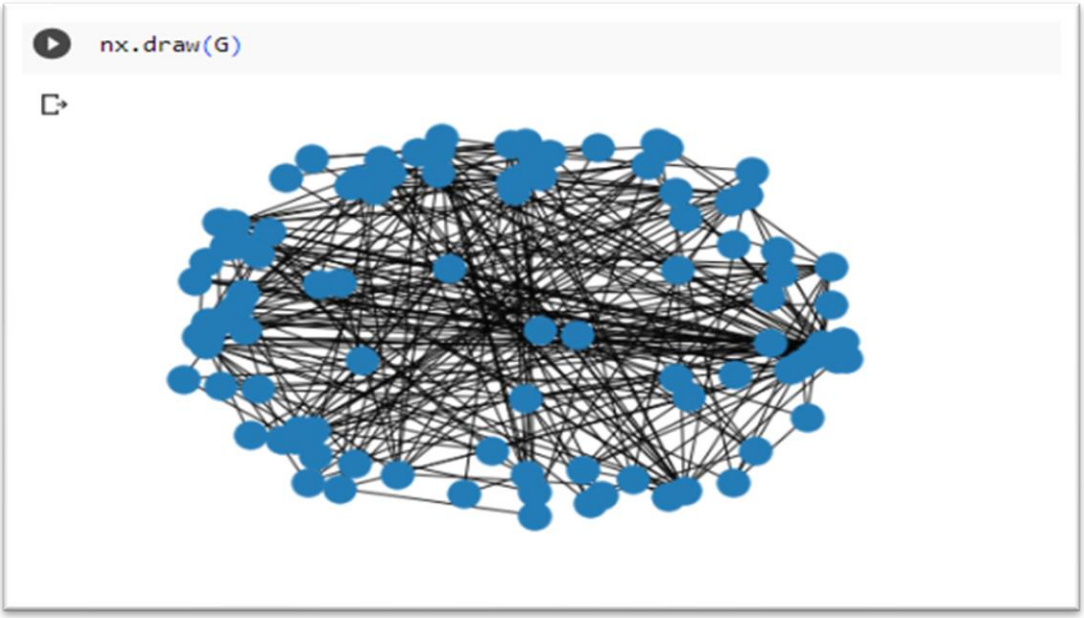
Şekil 2. 15. Eğitim veri setine ait kelimelerin sayısal gösterimi.

```
] X_train_vect_avg[3]

   array([ 0.94671834,  0.8898453 ,  0.02408131, -0.1579449 ,  0.4716353 ,
           0.8287916 , -0.11681191,  0.9226013 ,  0.28612295, -0.40770912,
          -0.4907464 ,  0.14708412, -0.22702017, -0.0414156 , -0.03553596,
           0.39375004,  0.28004533, -0.43600807,  0.18120793, -0.11134224,
           0.46238807, -0.33120903, -0.87942827,  0.1896541 , -0.67775905,
          -0.31222004, -0.6344403 ,  0.16578607, -0.02042372, -0.6026596 ,
           0.19202568, -0.30513003, -0.28187943, -0.7501682 ,  0.09945751,
          -0.37142748,  0.47179532, -0.03064225, -0.31239495,  0.04658081,
           0.3680356 ,  0.25823113, -0.11760823,  0.14153548,  0.22772491,
          -0.53326803, -0.21956143, -0.16265936, -0.12453497,  0.0347907 ,
          -0.08320802, -0.30477962,  0.520729  ,  0.13578667,  0.5227691 ,
           1.0471805 , -0.31736594,  0.3523534 , -0.12405057,  0.33648106,
          -0.00230091,  0.43985298, -1.1187885 ,  0.3483792 ,  0.04803265,
           0.4052241 ,  0.23234354,  0.34685084, -0.24290612, -0.31654862,
           0.22631645, -0.61593205,  0.06027976,  0.22029021,  0.09228443,
           0.02485234, -0.634375  ,  0.42518824,  0.6014244 ,  0.7869948 ,
           0.03642805,  0.59330285, -0.08601552, -0.07422362, -0.23358388,
           0.72414833, -0.5101947 ,  0.57780343, -0.22136046,  0.04737488,
           0.16971192,  0.27900007, -0.3489163 , -0.23185122,  0.3623814 ,
           0.13860518,  0.28434116,  0.07421261, -0.05792912, -0.27178353],
         dtype=float32)
```

Şekil 2. 16. Örnek bir dokümana ait 100 elemanlı ortalama matris değerleri.

Dönüşümden sonra Şekil 2.16' da görüldüğü gibi her bir doküman için elde edilen ortalama kelime gömme vektörleri üzerinden ayrıca yine her doküman için görünürlük grafı dönüşümü uygulanmaktadır. "Networkx" kütüphanesinde bulunan "NaturalVG" sınıfı yardımıyla bu zaman serisi grafiği ağ grafiğine dönüştürülür ve "G" olarak tanımladığımız değişkende saklanır. Şekil 2.17' de dönüştürülen matrisimize ait görünürlük graf modeli gösterilmektedir.



Şekil 2. 17. Kullanılan veri seti içerisindeki bir metin bloğuna ait görünürlük grafı.

2.6. METİN SINIFLANDIRMA YÖNTEMLERİ

Metin sınıflandırma, bir makine öğrenimi görev türüdür ve metinleri analiz ederek onları belirli etiketlerle ilişkilendirmektir. Metin sınıflandırma yöntemleri doğal dil işleme alanında yaygın olarak kullanılmaktadır. Naive Bayes sınıflandırıcı, SVM, KNN, YSA ve derin öğrenme yöntemleri metin sınıflandırma da kullanılan başlıca sınıflandırma metotlarıdır. Çalışmada sınıflandırma yöntemi olarak YSA ve derin öğrenme yöntemlerinden olan CNN modeli kullanılmaktadır.

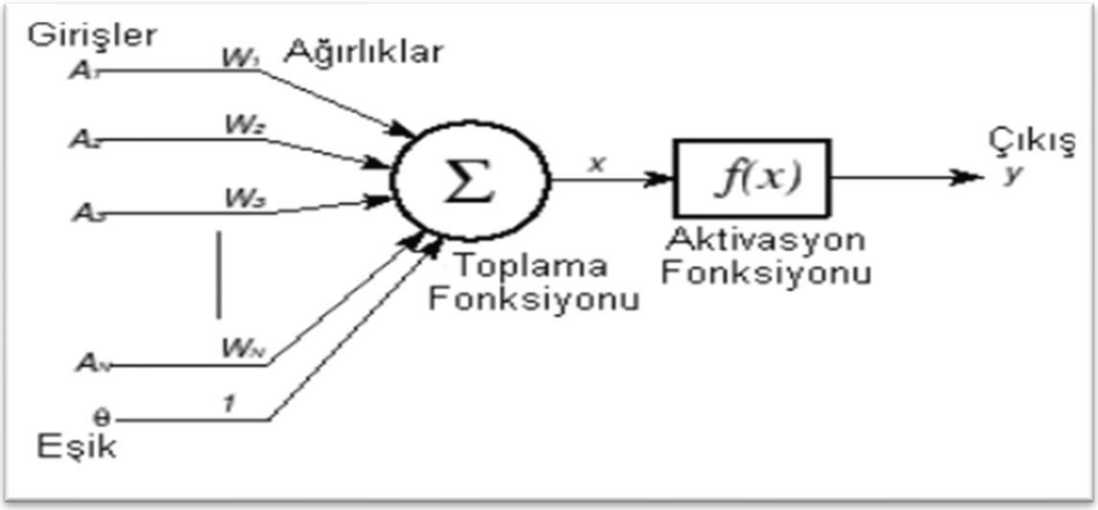
2.6.1. Yapay Sinir Ağları

Yapay sinir ağları, insan beyninin sinir hücrelerinden ilham alınarak tasarlanmış bir yapay zekâ modelidir. Yapay sinir ağları karmaşık veri ilişkilerini öğrenme ve tanıma yeteneğine sahiptir. Son yıllarda özellikle derin öğrenme yöntemleriyle birlikte büyük başarılar elde edilmektedir [69]. YSA, üç temel bileşenden oluşur. Bu bileşenler, girdi katmanı, gizli katmanlar ve çıktı katmanıdır. Girdi katmanı, veri setinden alınan girdilerin başlangıç katmanıdır. Bu katman verilerin özellik değerlerini alır ve sinir ağı içerisindeki gizli katmanlara iletir.

Gizli katman, girdi katmanı ve çıktı katmanı arasında kalan bir veya birden fazla katmandır. Her gizli katman, kendinden bir önceki katmandaki çıktıları alır. Bu çıktıları ağırlıklarla çarparak belirli bir aktivasyon fonksiyonuna gönderip yeni çıktılar üretir. Bu gizli katmanlar, veri setindeki karmaşık yapıları öğrenme ve özelliklerini çıkararak temsil etmede oldukça önemlidir.

Çıktı katmanı, sinir ağının son katmanıdır ve veri setine ait çıktıları sağlar. Bu katman, sınıflandırma, regresyon, dil oluşturma gibi istenen çıktı formatına dönüştürür. Şekil 2.18'de basit bir YSA mimarisi gösterilmektedir [70].

YSA, sınıflandırma, dil işleme, görüntü işleme gibi pek çok alanda oldukça başarılıdır. YSA, derin öğrenme tekniklerinin temeli olarak kullanılır. Özellikle derin öğrenme modelleriyle birlikte kullanıldıklarında, doğal dil işleme, görüntü tanıma, ses tanıma gibi alanlarda oldukça başarılı sonuçlar elde edebilirler.

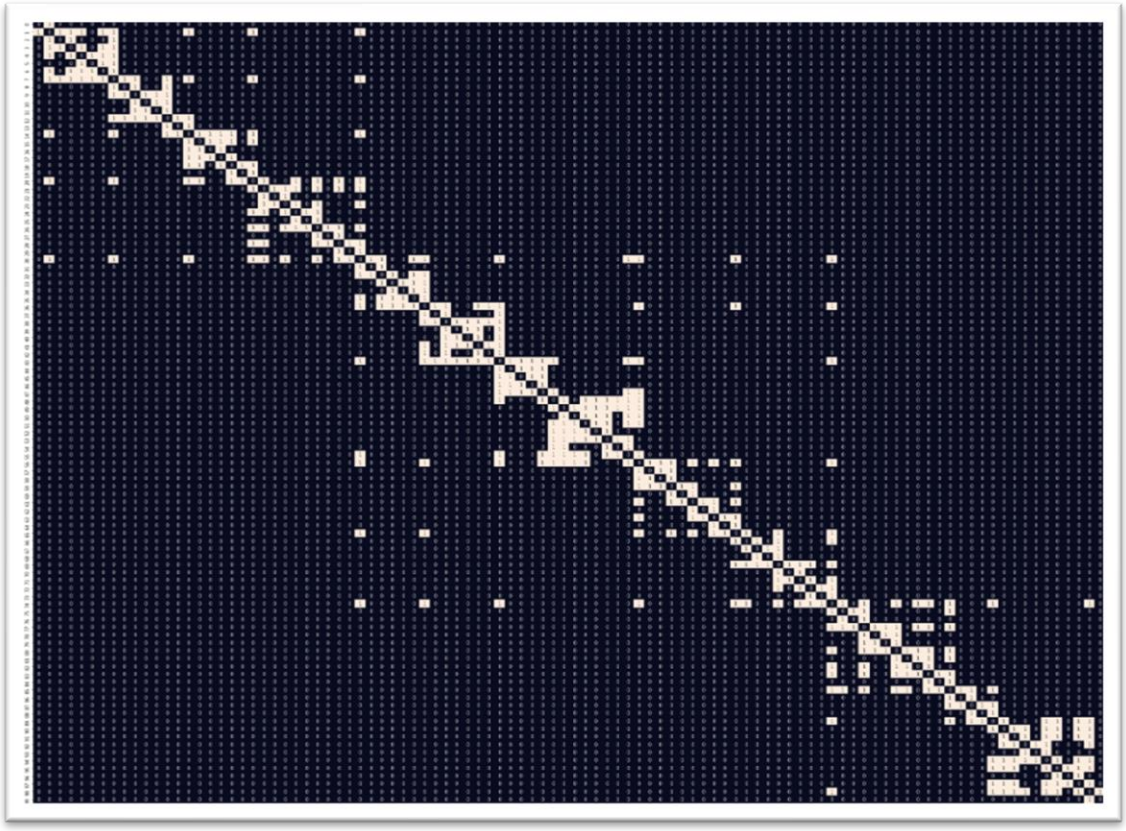


Şekil 2. 18. YSA mimarisi [69].

Ayrıca bu tez çalışmasında öne sürülen 2 boyutlu graf temsilli yapıyı tek boyuta indirgeyerek YSA' ya gönderilmektedir. Bu işlem için veriler tek boyutlu bir liste yapısana dönüştürülmektedir. Şekil 2.19' da bir örneği görülen komşuluk matrisleri modele girdi oluşturabilmek için verilerin tek boyutlu bir liste yapısına dönüştürülmesi gerekmektedir. Bu sebeple oluşturulan komşuluk matrislerine ait düğüm dereceleri hesaplanılarak, tek boyutlu liste haline getirilmektedir. Oluşturulan düğüm derece listelerini [0,1] aralığında normalize edilip yapay sinir ağı eğitiminde kullanılmaktadır. Bu değerler Şekil 2.20' de gösterilmektedir.

Bu tez çalışmasında uygulanan bir diğer yöntem ise Şekil 2.19' da görülen komşuluk matrisine ait (100*100 boyut için verilen bir örnek) ana diagonal ait veriler de sinir ağına gönderilip başarısı test edilmiştir.

Çalışmada girdi olarak verilen özellik sayısına göre klasik kelime gömme yöntemlerini sıralamak için bir yapay sinir ağı modeli oluşturulmaktadır. Şekil 2.21 ve Şekil 2.22'de kullanılan YSA model ve yapısı gösterilmektedir.



Şekil 2. 19. 100x100 (2D) boyutlu bir komşuluk matrisi örneği.

```
In [21]: normalized_degrees

Out[21]: array([0.        , 0.47058824, 0.11764706, 0.23529412, 0.23529412,
                0.11764706, 0.23529412, 0.64705882, 0.11764706, 0.23529412,
                0.11764706, 0.11764706, 0.35294118, 0.05882353, 0.52941176,
                0.23529412, 0.11764706, 0.17647059, 0.23529412, 0.05882353,
                0.58823529, 0.35294118, 0.05882353, 0.23529412, 0.23529412,
                0.05882353, 0.41176471, 0.05882353, 0.29411765, 0.05882353,
                1.        , 0.05882353, 0.23529412, 0.17647059, 0.11764706,
                0.23529412, 0.70588235, 0.11764706, 0.35294118, 0.17647059,
                0.17647059, 0.29411765, 0.17647059, 0.88235294, 0.17647059,
                0.17647059, 0.17647059, 0.35294118, 0.47058824, 0.41176471,
                0.29411765, 0.35294118, 0.23529412, 0.35294118, 0.17647059,
                0.52941176, 0.88235294, 0.11764706, 0.11764706, 0.29411765,
                0.05882353, 0.29411765, 0.05882353, 0.23529412, 0.05882353,
                0.52941176, 0.17647059, 0.11764706, 0.05882353, 0.41176471,
                0.05882353, 0.17647059, 0.17647059, 0.05882353, 1.        ,
                0.17647059, 0.05882353, 0.47058824, 0.11764706, 0.11764706,
                0.41176471, 0.17647059, 0.23529412, 0.35294118, 0.05882353,
                0.52941176, 0.05882353, 0.17647059, 0.05882353, 0.47058824,
                0.35294118, 0.29411765, 0.23529412, 0.05882353, 0.41176471,
                0.35294118, 0.05882353, 0.35294118, 0.47058824, 0.        ])
```

Şekil 2. 20. Komşuluk matrisi düğüm derece listesine ait normalizasyon değerleri(1D).

```python
model2 = tf.keras.Sequential([
    tf.keras.layers.Input((300,)),
    tf.keras.layers.Dense(64, activation='relu'),
    tf.keras.layers.Dense(64, activation='relu'),
    tf.keras.layers.Dense(64, activation='relu'),
    tf.keras.layers.Dense(64, activation='relu'),
    tf.keras.layers.Dense(64, activation='relu'),
    tf.keras.layers.Dense(64, activation='relu'),
    tf.keras.layers.Dropout(0.3),
    tf.keras.layers.Dense(len(set(df['label'])), activation='softmax')
])

model2.compile(
    loss='sparse_categorical_crossentropy',
    optimizer='adam',
    metrics=['accuracy']
)

model2.summary()
```

Şekil 2. 21. Çalışmada kullanılan YSA modeli.

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_2 (Dense)             (None, 64)                19264

 dense_3 (Dense)             (None, 64)                4160

 dense_4 (Dense)             (None, 64)                4160

 dense_5 (Dense)             (None, 64)                4160

 dense_6 (Dense)             (None, 64)                4160

 dense_7 (Dense)             (None, 64)                4160

 dropout_1 (Dropout)         (None, 64)                0

 dense_8 (Dense)             (None, 4)                 260

=================================================================
Total params: 40,324
Trainable params: 40,324
Non-trainable params: 0
```

Şekil 2. 22. Çalışmada kullanılan YSA model yapısı.

2.6.2. Derin Öğrenme

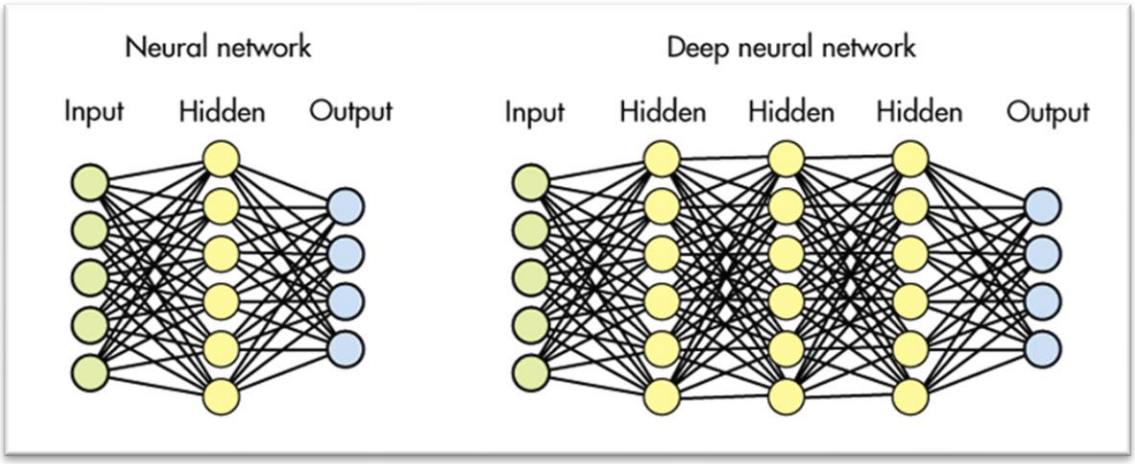
Derin öğrenme (deep learning), YSA temelli bir makine öğrenme yöntemidir. Bu yöntem büyük veri kümelerini analiz etmek için kullanılır. Çok katmanlı ve karmaşık yapıları sayesinde yüksek seviyede özellik çıkarımı ve temsil öğrenimi yapma yeteneğine sahiptir. Derin öğrenme yöntemleri, ağırlıkları ve biasları otomatik olarak ayarlamak için geriye yayılım (backpropagation) adı verilen bir optimizasyon yöntemi kullanır. Bu yöntem, eğitim veri seti üzerindeki tahmin hatalarını en aza indirgeyerek ağın performansını iyileştirir. Nesne tespiti, yüz tanıma, görüntü sınıflandırma, metin sınıflandırma, dil çevirisi, metin üretimi, ses tanıma gibi çeşitli uygulamalarda kullanılabilir.

Sinir ağları, fotoğraflar veya ses gibi gözlemsel verileri anlamlandırmak için, verileri birbirine bağlı düğüm katmanlarından geçirir. Bilgi bir katmandan geçtiğinde, o katmandaki her bir düğüm, veriler üzerinde basit işlemler gerçekleştirir ve sonuçları seçerek diğer düğümlere iletir. Sonraki her katman, ağ çıktıyı oluşturana kadar bir öncekinden daha yüksek düzeyde bir özelliğe odaklanır. Derin öğrenme modelleri, büyük miktarda veriyi eğitmek için çok zaman harcar, bu nedenle yüksek performanslı hesaplama çok önemlidir [71].

Derin öğrenmenin en büyük avantajlarından biri, sinir ağlarının verilerden, daha önce görünür olmayan gizli içgörüleri ve ilişkileri ortaya çıkarmak için kullanılmasıdır. Büyük ve karmaşık verileri analiz edebilen daha sağlam makine öğrenimi modelleriyle şirketler, dolandırıcılık tespitini, tedarik zinciri yönetimini ve siber güvenliği iyileştirebilir. Derin öğrenme algoritmaları, değerli iş ve müşteri içgörüleri sağlamak için sosyal medya gönderilerini, haberleri ve anketleri analiz ederek metin verilerine bakma üzere eğitilebilir. Derin öğrenme algoritması, insanların ham verilerden manuel olarak özellikleri seçip çıkartmasını gerektirmediği için zamandan tasarruf sağlayabilir. Bir derin öğrenme algoritması uygun şekilde eğitildiğinde, insanlardan daha hızlı bir şekilde binlerce görevi tekrar tekrar gerçekleştirebilir. Derin öğrenmede kullanılan sinir ağları birçok farklı veri tipine ve uygulama yazılımına uygulanabilme özelliğine sahiptir. Ek olarak, derin öğrenme modeli, yeni verilerle yeniden eğitilerek

farklı durumlar için uygunlaştırılabilme özelliğine sahiptir. Şekil 2.21'de YSA ve derin öğrenme mimarileri gösterilmektedir [72].

Evrişimli sinir ağları (CNN), Recurrent sinir ağları (RNN), Uzun kısa vadeli bellek (LSTM) başlıca derin öğrenme yöntemleridir. Bu çalışmada graf temsilli yöntem YSA ve CNN derin öğrenme yöntemi kullanılarak iki farklı yöntem ile sıralanmış ve algoritmaların model üzerindeki etkisi incelenmiştir.



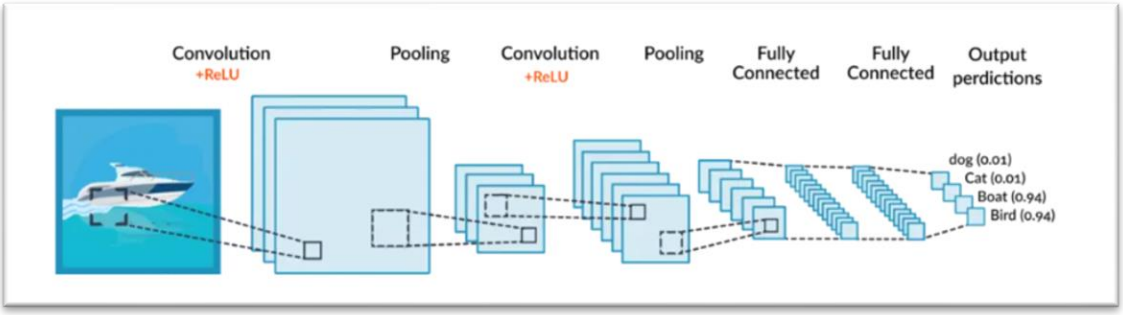
Şekil 2. 23. Yapay sinir ağı ve derin yapay sinir ağları [73].

Evrişimli Sinir Ağları

Evrişimli sinir ağları (CNN), genellikle görüntüler ve videolar gibi görsel verilerin işlenmesi ve analiz edilmesi için kullanılan bir derin öğrenme algoritmasıdır.

Veri soyutlamaları ve katmanlar arasındaki temsil hiyerarşisi ile bağlantılı yavaş öğrenme süreci nedeniyle, bazı derin öğrenme algoritmaları yüksek boyutlu görüntü verilerini kullanırken yüksek hesaplama maliyeti eşlik edebilir. CNN ise büyük boyutlardaki veriyi önemli ölçüde ölçeklendirir CNN mimarisinin genel amacı, öğrenme kapasitesini kaybetmeden parametreleri azaltmaktır. Farklı operasyonlarla görsellerdeki özellikleri yakalayan ve onları sınıflandıran bu algoritma farklı katmanlardan oluşmaktadır. CNN modelleri oluştururken, direkt olarak ham veri işlendiğinden klasik makine öğrenmesi algoritmalarına kıyasla veri ön işleme

kısmında fazla iş yükü yoktur. Şekil 2.22'de CNN mimarisi gösterilmektedir. CNN mimarisinin temel bileşenleri şu şekildedir:



Şekil 2. 22. CNN mimarisi [53].

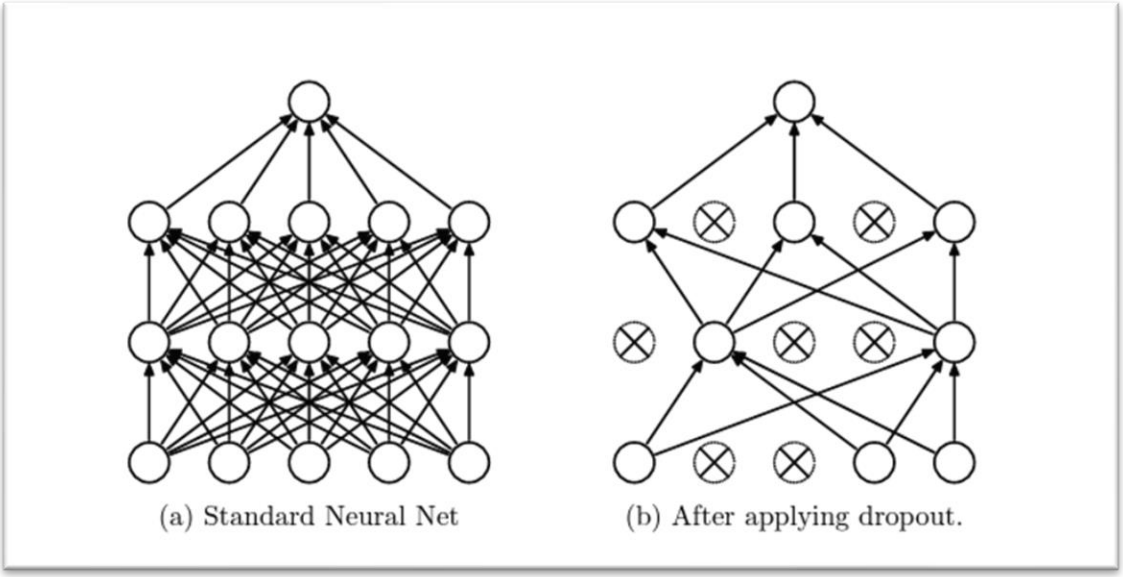
- ***Evrişim Katmanları (Convolutional Layers):*** Bu katman giriş katmanından gelen verilerin ayırt edici özelliklerini kullanarak öğrenmeyi amaçlamaktadır. Bu işlem özellikleri vurgulayan yeni bir özellik haritası oluşturur. Birden fazla filtre kullanılarak, farklı özellik haritaları oluşturulur.

- ***Havuzlama Katmanları (Pooling***): Evrişim sonrası elde edilen özellik haritalarının boyutunu azaltmak için kullanılır. Bu katmanda öğrenme işlemi yoktur fakat hesaplama karmaşıklığını azaltmak için parametrelerden bir kısmı çıkarılır. Maksimum, ortalama ve minimum olmak üzere üç çeşit havuzlama yöntemi bulunmaktadır. Bu işlemler ağın öğrenme sürecini hızlandırır ve aşırı öğrenmeyi (overfitting) engeller [53].

- ***Tam Bağlı Katmanlar (Fully Connected Layers):*** Evrişim ve havuzlama katmanlarının ardından, ağda sıkıştırılmış bilgiler elde edilir ve bunlar tam bağlı katmanlara iletilir. Bu katmanlar sınıflandırma veya regresyon gibi sonuçları üretmek için kullanılırlar. Tam bağlantılı bir katmanda, son katman düzleştirilir ve çıktı tek boyutlu bir vektöre dönüştürülür. Tam bağlı katmanlar birden çok katmana sahip olabilir. Bu katmanın son katmandaki nöron sayısı, veri setine ait sınıf sayısı ile eşit olması önemlidir [53].

- ***Seyreltme katmanı (Dropout):*** Srivastava ve arkadaşları tarafından önerilen bu katman derin öğrenme yöntemlerinde aşırı uyumu (overfitting) önlemek

amacıyla kullanılır. Dropout katmanı, sinir ağı eğitimi sırasında rastgele seçilen bazı nöronları devre dışı bırakarak modelin genelleştirmesine yardımcı olur. Literatürde genellikle dropout işleminin %20 ile %50 arasında bir oran ile yapıldığı görülmektedir. Seyreltme işlemi özellikle büyük ve karmaşık veri kümelerinde, karmaşık sinir ağlarında kullanıldığında etkilidir. Uygun dropout oranının seçimi modelin performansı için önemlidir [74]. Çalışmada dropout oranı 0.3 olarak seçilmiştir. Şekil 2.23'te standart bir sinir ağı ile seyreltme işlemi uygulanmış bir sinir ağı karşılaştırmalı olarak gösterilmektedir.



Şekil 2. 23. Standart bir sinir ağı ve dropout işlemi sonrası sinir ağı örneği [74].

- ***Aktivasyon Fonksiyonları:*** Aktivasyon fonksiyonları, sinir ağlarında her bir nöron çıkışına uygulanan maatematiksel işlemlerdir. Bu fonksiyonlar, sinir ağlarının lineer olmayan işlemleri öğenmesini ve karmaşık bağımlılıkları ele almasını sağlar. Aktivasyon fonksiyonları, geri yayılım (backpropagation) algoritmasını da kullanarak ağırlıkların güncellenmesini sağlar [75]. En yaygın kullanılan aktivasyon fonksiyonları, Sigmoid, ReLu (Rectified Linear Unit), Leaky ReLu, Tanh ve Softmax fonksiyonlarıdır. Bu çalışmada ReLu ve Softmax aktivasyon fonksiyonları kullanılmıştır.

CNN mimarisinin uyguladığı bu katmanlar ve filtreler, matrislerdeki özellik çıkarım başarısında oldukça etkilidir. Bu başarı, metinlerin sayısallaştırılarak matrislere dönüştürülmesi sonrası sınıflandırılmasında da kendini göstermektedir. Bu sebeple graf temsilli yöntemle oluşturulan 2 boyutlu girdi verileri tek boyuta indirgenerek CNN mimarisi ile sınıflandırılmış ve CNN'in özellik çıkarım performansı ile çalışmada önerilen graf temsilli öğrenme birlikte kullanıldığında, diğer geleneksel kelime gömme yöntemlerine kıyasla başarılı bir sonuç göstermiştir. Çalışmada kullanılan CNN mimari modeli Şekil 2.24'te gösterilmektedir. Model incelendiğinde, giriş katmanına 10000 boyutlu bir vektör gönderilmektedir, sonrasında bir boyutlu evrişimli sinir ağları kullanabilmek için yeniden şekillendirme katmanı kullanılmıştır. 64 filtre ile 6 adet evrişimli katman kullanılmaktadır. Çalışmanın evrişimli katman bölümünde "ReLu" aktivasyon fonksiyonları kullanılmaktadır. Bu sayede verinin belirli özelliklerini vurgulayan yeni özellik haritaları oluşturulmaktadır. Aşırı öğrenmeyi (overfitting) engellemek amacıyla 0.3 oranında dropout katmanı eklenilmiştir. Bu işlemlerden sonra düzleştirme (flatten) katmanıyla özellik haritaları düzleştirilerek tek bir vektör halini almaktadır. Modele son olarak, verileri sınıflandırabilmek için tam bağlı katman eklenilmektedir. Bu kısımda etiket sayısı kadar çıkış birimi bulunan "softmax" aktivasyon fonksiyonu kullanılmaktadır. Şekil 2.25'te modelin katmanlarına ait çıktı şekilleri gösterilmektedir. Buna göre tam bağlı katmanda (dense) sınıf sayısı kadar yani 4 çıkış biriminin olduğu görülmektedir. Toplamda modelde 9856260 adet parametre kullanıldığı da görülmektedir. Bu parametreler, sinir ağının öğrenme sürecinde optimize edilir ve modelin doğru sınıflandırma yapmasında etkilidir.

```python
import tensorflow as tf

model2 = tf.keras.Sequential([
    tf.keras.layers.Input((100, 100, 1)),
    tf.keras.layers.Conv2D(64, (3, 3), activation='relu'),
    tf.keras.layers.MaxPooling2D((2, 2)),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(64, activation='relu'),
    tf.keras.layers.Dense(64, activation='relu'),
    tf.keras.layers.Dense(64, activation='relu'),
    tf.keras.layers.Dense(64, activation='relu'),
    tf.keras.layers.Dropout(0.5),
    tf.keras.layers.Dense(64, activation='relu'),
    tf.keras.layers.Dense(64, activation='relu'),
    tf.keras.layers.Dropout(0.3),
    tf.keras.layers.Dense(len(set(df['label'])), activation='softmax')
])

model2.compile(
    loss='sparse_categorical_crossentropy',
    optimizer='adam',
    metrics=['accuracy']
)

model2.summary()
```

Şekil 2. 24. Çalışmaya ait CNN modeli.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 98, 98, 64)        640

 max_pooling2d (MaxPooling2D  (None, 49, 49, 64)       0
 )

 flatten (Flatten)           (None, 153664)            0

 dense (Dense)               (None, 64)                9834560

 dense_1 (Dense)             (None, 64)                4160

 dense_2 (Dense)             (None, 64)                4160

 dense_3 (Dense)             (None, 64)                4160

 dropout (Dropout)           (None, 64)                0

 dense_4 (Dense)             (None, 64)                4160

 dense_5 (Dense)             (None, 64)                4160

 dropout_1 (Dropout)         (None, 64)                0

 dense_6 (Dense)             (None, 4)                 260

=================================================================
Total params: 9,856,260
Trainable params: 9,856,260
Non-trainable params: 0
_________________________________________________________________
```

Şekil 2. 25. Çalışmada kullanılan CNN model yapısı.

2.8. MODEL BAŞARIM ÖLÇÜTLERİ

Sınıflandırıcı performanslarını değerlendirmek için doğruluk ve F-Skor gibi yaygın metrikler kullanılır. Bu metrikler, Çizelge 2.3' teki gibi ifade edilen bir karmaşıklık matrisinde açıklanan doğru veya yanlış sınıflandırılmış pozitif ve negatif örneklerin sayıları kullanılarak hesaplanır [18].

Karmaşıklık Matrisi

Karmaşıklık matrisi, en az 2 farklı sınıfa ait modellerin sonuçlarını analiz edebilmek için kullanılan bir yöntemdir. Bu 2x2 boyutundaki matris içerisinde yatay hiza gerçek değerleri, dikey hiza ise tahmin edilen değerleri temsil etmektedir. Çizelge 2.3’ de iki sınıflı bir model için üretilmiş temsili bir karmaşıklık matrisi gösterilmiştir. Burada

DP doğru pozitif, DN doğru negatif, YP yanlış pozitif, YN ise yanlış negatif olarak adlandırılır.

Çizelge 2. 3. İkili sınıflandırma için karmaşıklık matrisi.

		Gerçek Değer	
		Pozitif (P)	Negatif (N)
Tahmin Edilen	Pozitif (P)	DP (Doğru Pozitif)	YP (Yanlış Pozitif)
	Negatif (N)	FP (Yanlış Negatif)	DN (Doğru Negatif)

Örneğin; X, Y, Z ve T adında 4 adet sınıf olduğu düşünülürse, bu durumda hayali karmaşıklık matrisi Çizelge 2.4' deki gibi gösterilebilir. Burada mavi renkler doğru sınıflandırılan, gri renkler ise yanlış sınıflandırılan sonuçları temsil etmektedir.

Çizelge 2. 4. Dört sınıflı karmaşıklık matrisine örnek.

	X	**Y**	**Z**	**T**
X	15	3	4	6
Y	6	14	2	8
Z	3	4	18	2
T	2	3	4	20

- ***Doğruluk-Hata Oranı:*** Doğruluk oranı, model performansını ölçmek için en belirgin ve basit yöntemdir. Doğru sınıflandırılmış numune sayısının (DP + DN) toplam numune sayısına (DP + DN + YP + YN) oranı olarak tanımlanır. Hata oranı, yanlış sınıflandırılmış numune sayısının (YP + YN) toplam numune sayısına (DP + DN + YP + YN) oranı olarak tanımlanır [18]. Denklemler 2.3 ve 2.4' te gösterilmiştir.

$$\text{Doğruluk} = \frac{\text{DP} + \text{DN}}{\text{DP} + \text{YP} + \text{YN} + \text{DN}} \tag{2.3}$$

$$\text{Hata Oranı} = \frac{\text{YP} + \text{YN}}{\text{DP} + \text{YP} + \text{YN} + \text{DN}} \tag{2.4}$$

- ***Kesinlik:*** Kesinlik, doğru tahmin edilen örneklerin toplam pozitif örnek sayısına oranı olarak tanımlanır. İfadeye ait denklem 2.5' te gösterilmiştir [18].

$$\text{Kesinlik} = \frac{DP}{\text{DP} + \text{YP}} \tag{2.5}$$

- ***Duyarlılık:*** Duyarlılık doğru sınıflandırılmış pozitif numunelerin toplam pozitif numune sayısına oranını tanımlar. Denklem 2.6' da verilen formül ile hesaplanabilir.

$$\text{Duyarlılık} = \frac{DP}{\text{DP} + \text{YN}} \tag{2.6}$$

- ***F-Skor:*** Kesinlik ve duyarlılık kriterleri tek başına anlamlı bir karşılaştırma yapmak için yetersizdir. Her iki kriteri bir araya getirmek daha doğru bilgi sağlar. F-Skor bu amaçla tanımlanır. Duyarlılık ve kesinliğin harmonik ortalamasına F-Skor denir ve denklem 2.7 ile hesaplanır [19].

$$\text{F} - \text{Skor} \quad = \frac{2xDuyarlılıkxKesinlik}{Duyarlılık + Kesinlik} \tag{2.7}$$

BÖLÜM 3

DENEYSEL ÇALIŞMALAR VE TARTIŞMA

Çalışmada literatürde temsil uzayı için en çok kullanılan kelime gömme yöntemleri olan WordeVec (CBOW, Skip-Gram), FastText, BERT ve Glove algoritmaları kullanılmıştır. Geliştirme ortamı olarak Jupyter Notebook içerisindeki Python Dili kullanılmaktadır. İşleme ve öğrenme görevleri için "Scikit-learn", "nltk", "tensorflow", "numpy", "pandas" ve "gensim" kütüphaneleri kullanılmaktadır. Graf temsilli yaklaşımda "ts2vg" ve "networkx" kütüphanelerinden, görsel grafikler için ise "seaborn" kütüphanesinden faydalanılmıştır.

Öznitelik (kelime) sayısının sınıflandırma sonucuna olan etkisini inceleyebilmek için bu çalışmada 100 ile 1500 aralığında değişen kelimeler ki-kare metriği ile seçilerek bu öznitelik setleri ile deneyler gerçekleştirilmiştir. Bir kelimenin her sınıf için normalleştirilmiş kullanımıyla beraber bu sayıların sınıflar arası sapmaları ki-kare metriği ile hesaplanmış ve sınıflar arası standart sapması en yüksek kelimeler öznitelik olarak seçilmiştir.

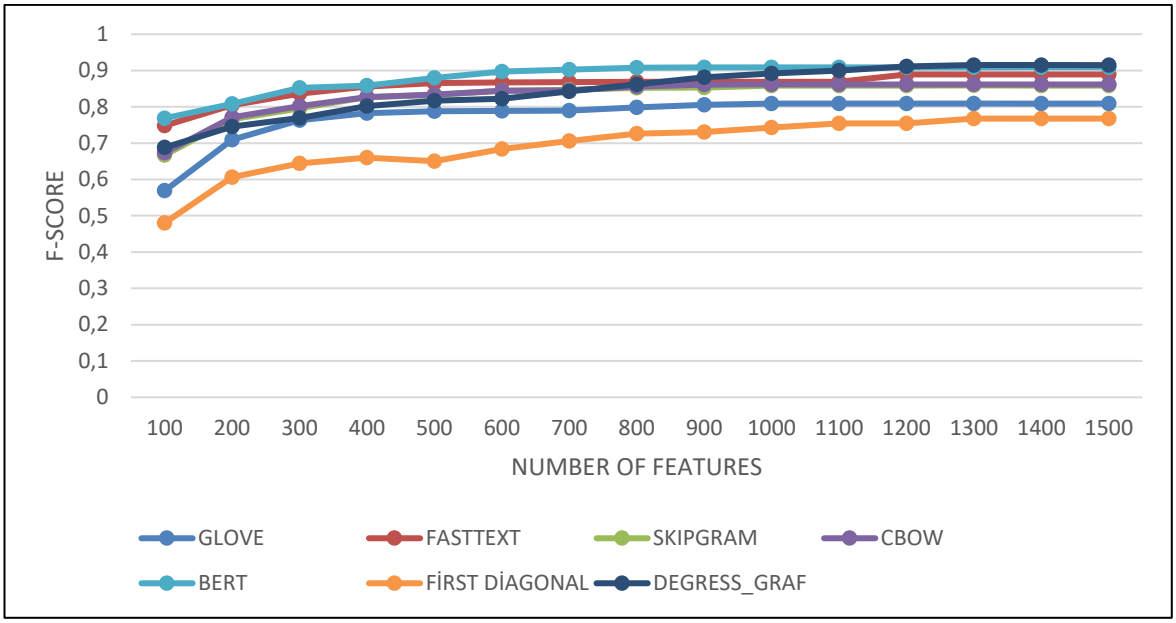
Klasik kelime yöntemleri için çalışmaya ait sonuçlar, verilen kelime gömme yöntemleri ve her yöntem için çeşitli sayıda öznitelik için Çizelge 2.4'te verilmiştir. Klasik kelime yöntemleri sınıflandırması için bir yapay sinir ağı modeli oluşturulmuştur. Sonuçlar, 600 kelimeye kadar BERT in daha başarılı, 600 kelime sonrası özellik seçiminde graf temsilli yöntemin daha başarılı olduğunu göstermektedir. Bu başarıyı Fasttex, CBOW, ve Skip-Gram ve GloVe modelleri takip etmektedir. 1500 kelime için graf temsilli yöntem %91.2, BERT, %90.91, Fasttext %88.91, Skip-Gram %85.86, CBOW %86.15, GloVe %0,809 baları göstermiştir. Matrisin ana diagonalini girdi olarak verdiğimiz yaklaşım %76.72 ile diğer yöntemlere karşı rekabetçi olamamıştır. Sonuçlar ayrıca, en iyi puanların 1500 kelimelik bir özellik seti için elde edildiğini gösterirken, makul sayıda özellik kullanımı ile (~500)

tüm kelime gömme teknikleri için en iyi sonuçlar ile karşılaştırılabilir sınıflandırma doğrulukları sağladığını ortaya koymuştur. Şekil 3.1 ve Şekil 3.2 de doğruluk ve F-skor açısından grafiksel dönüşümleri gösterilmektedir.

Yine Çizelge 3.1 ve Çizelge 3.2' de tüm sonuçlar gösterilmektedir.

Çizelge 3. 1. Ki-kare özellik sayısına göre doğruluk açısından sınıflandırma sonuçları.

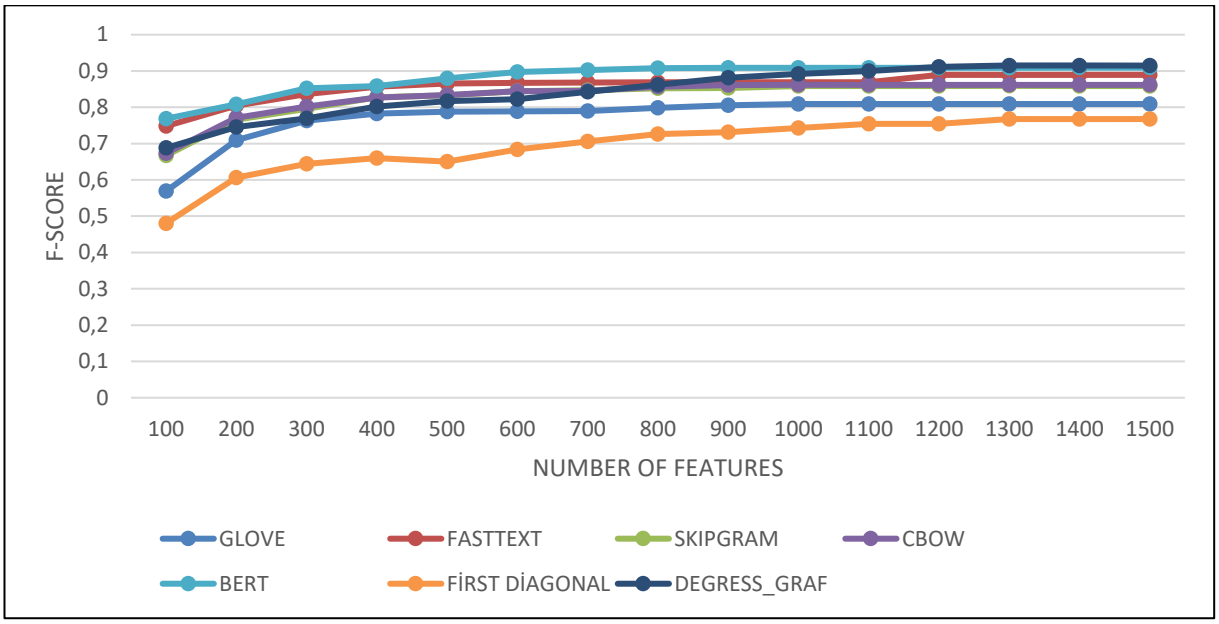
KELİME SAYISI	GLOVE	FASTTEXT	SKIPGRAM	CBOW	BERT	FİRST_DIAGONAL	DEGRESS_GRAF
100	0,5688	0,7482	0,6688	0,675	0,7685	0,498	0,6886
200	0,7092	0,8046	0,7687	0,7721	0,8082	0,6061	0,745
300	0,7632	0,8364	0,797	0,8024	0,8524	0,6457	0,8022
400	0,7825	0,8562	0,8282	0,827	0,8584	0,6601	0,8422
500	0,7876	0,8654	0,8323	0,8339	0,8796	0,6602	0,8727
600	0,7885	0,8673	0,8441	0,8444	0,8948	0,6832	0,8931
700	0,7896	0,8682	0,8477	0,8455	0,9022	0,7021	0,9068
800	0,7982	0,8687	0,852	0,8562	0,9076	0,7267	0,9112
900	0,8058	0,869	0,8533	0,8616	0,9085	0,7319	0,9117
1000	0,8087	0,8691	0,8585	0,8615	0,9088	0,7426	0,9124
1100	0,8088	0,889	0,8586	0,8617	0,9089	0,7543	0,912
1200	0,8088	0,8892	0,8588	0,8617	0,9089	0,7543	0,912
1300	0,8089	0,8893	0,859	0,8617	0,909	0,7672	0,9121
1400	0,809	0,8893	0,8588	0,8618	0,9091	0,7673	0,9121
1500	0,809	0,8891	0,8586	0,86158	0,9091	0,7672	0,912



Şekil 3. 1. Tüm modellere ait accuracy açısından grafiksel gösterim.

Çizelge 3. 2. Ki-kare özellik sayısına göre f-skor açısından sınıflandırma sonuçları.

KELİME SAYISI	GLOVE	FASTTEXT	SKIPGRAM	CBOW	BERT	FİRST_DIAGONAL	DEGRESS_GRAF
100	0,5689	0,7481	0,6671	0,6735	0,7685	0,4804	0,688
200	0,7093	0,8045	0,7654	0,7711	0,8081	0,6058	0,7452
300	0,7632	0,8363	0,7958	0,8022	0,8523	0,6445	0,7696
400	0,7825	0,8562	0,8271	0,8264	0,8585	0,6598	0,8019
500	0,7876	0,8654	0,8326	0,8336	0,8796	0,6501	0,8169
600	0,7885	0,8672	0,8442	0,8443	0,8974	0,684	0,8225
700	0,7895	0,8682	0,8472	0,8453	0,9023	0,7056	0,8424
800	0,7982	0,8687	0,852	0,8562	0,9077	0,7258	0,8622
900	0,8057	0,8691	0,8533	0,8611	0,9085	0,731	0,8816
1000	0,8086	0,8691	0,8582	0,8615	0,9088	0,7432	0,8916
1100	0,8087	0,869	0,8583	0,8616	0,9089	0,7547	0,9089
1200	0,8087	0,8891	0,8584	0,8616	0,9089	0,7546	0,911
1300	0,8088	0,8892	0,8589	0,8616	0,909	0,7674	0,9152
1400	0,8089	0,8892	0,8584	0,8616	0,9091	0,7675	0,9152
1500	0,8089	0,8891	0,8585	0,8616	0,9091	0,7674	0,9151

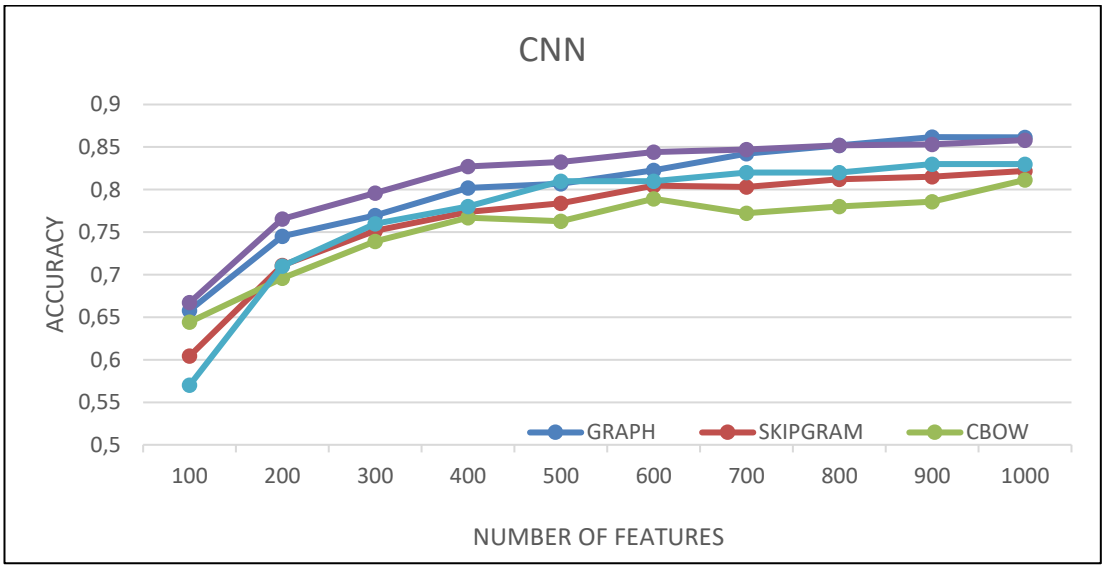


Şekil 3. 2. Tüm modellere ait f-skor açısından grafiksel gösterim.

Ayrıca 1500 kelimeye kadar olan bu sonuçlar, daha iyi okunabilirlik sağlamak için Çizelge 3.1'deki gibi çizilmiştir. Şekil 3.1'de grafiksel olarak gösterilmektedir. BERT, mevcut tüm özellik sayısı için tüm gömme yöntemlerinden daha iyi performans gösterildiği görülmektedir. Ancak, 500'e kadar olan düşük özellik sayısı için FastText, BERT ile benzer sonuçlar vermektedir. Aynı ailenin temsilcileri olan CBOW ve Skip-Gram, FastText ve BERT' in altında ve GloVe' nin üzerinde benzer sınıflandırma başarısı göstermektedir.

Çizelge 3. 3. Ki-kare özellik sayısına göre doğruluk açısından CNN sınıflandırma sonuçları.

KELİME SAYISI	GRAF	SKIPGRAM	CBOW	FASTTEXT	GLOVE
100	0,658	0,6043	0,6443	0,6671	0,5712
200	0,7452	0,711	0,6957	0,7654	0,7186
300	0,7696	0,7514	0,7392	0,7958	0,7675
400	0,8019	0,7738	0,767	0,8271	0,7872
500	0,8069	0,784	0,7627	0,8326	0,8156
600	0,8225	0,8047	0,789	0,8442	0,8189
700	0,8424	0,8031	0,7722	0,8472	0,8251
800	0,8522	0,812	0,7803	0,852	0,8262
900	0,8617	0,8151	0,7858	0,8533	0,8301
1000	0,8616	0,822	0,8114	0,8582	0,8312



Şekil 3. 3. CNN sonuçlarına ait grafiksel gösterim.

Önerilen graf temsilli öğrenme farklı ki-kare sayılarının model başarısı üzerindeki etkisini gözlemlerken, diğer geleneksel yöntemlerle birlikte sınıflandırılması ve karşılaştırılması hedeflenmiştir. Graf temsilli öğrenme modeline içerisinde bulundurduğu matris yapısı gereği YSA yerine CNN derin öğrenme yönteminin daha uygun olduğu gözlemlenmiştir. Bu sebeple graf temsilli öğrenmede sınıflandırma yöntemi olarak CNN kullanılmıştır. Diğer geleneksel yöntemlerle eşit şartlarda bir

karşılaştırma yapabilmek için CBOW, Skip-Gram, Glove, FastText ve BERT algoritması ile 1300 kelimeye kadar bir sınıflandırma yapılmıştır. Sınıflandırmada 1000 kelime için; %86.20 oranla önerdiğimiz graf temsili yöntem en başarılı olurken, %85.85 oranla FastText algoritması takip etmektedir. Bu algoritmaları Skip-Gram ve CBOW algoritmaları takip etmektedir. Çalışmada BERT ve Glove yöntemi sınıflandırma başarısı olarak %50' nin altında kalarak diğer yöntemler ile rekabet edemeyecek düzeyde olduğu gözlemlenmiştir. Buradan CNN mimari yapısının BERT ve Glove kelime gömme yöntemlerine uygun olmadığı gözlemlenilmektedir. Çizelge 3.2' de tüm kelime sayılarına ait sonuçlar gösterilmektedir. Şekil 2.3'te sonuçlara ait grafik eklenmiştir.

BÖLÜM 4

SONUÇLAR VE ÖNERİLER

Geleneksel kelime gömme yöntemleri, değişken kelime sayılarına sahip özellik kümeleriyle bir metin sınıflandırma görevi için test edilmiştir. Çalışma ilk olarak geleneksel kelime yöntemleri arasında bir karşılaştırma yapmıştır. Ki-kare yöntemi ile elde ettiğimiz veri setine ait en etkili 1500 kelime sırası ile 100-200-300…1500 şeklinde kelime gömme vektörlerine verilmiş, YSA modeli ile yapılan sınıflandırmada en başarılı sonucu BERT algoritması elde etmiş ve bu algoritmayı FastText yöntemi takip etmiştir. Beş kelime gömme yöntemi arasından (CBOW, Skip-Gram, Glove, FastText, BERT), BERT baskın olarak hepsinden daha iyi performans gösterirken, FastText 500 kelimeye kadar olan düşük özellik alanı için karşılaştırılabilir sonuçlar verebilmiştir. İlk iki yöntemden %2-3 daha düşük başarı düzeyine sahip olan Word2Vec modeli, daha büyük özellik kümeleri için doğruluğu artırma yeteneğine sahipken, FastText 500 kelimeden sonra doygunluğa ulaşmaktadır. GloVe, verilen görev için diğer gömme yöntemlerine göre %5-10 daha düşük sınıflandırma performansı gösterirken bu sonuçlar bu modelin diğerleriyle rekabetten uzak olduğunu ortaya koymaktadır. Ayrıca önerdiğimiz görünürlük graf dönüşümlü sınıflandırma modelinin, geleneksel kelime gömme modellerine göre başarıyı arttırdığı gözlemlenmiştir.

Çalışmamız çeşitli geleneksel kelime gömme modellerinin değişken öznitelik sayısı altındaki performanslarını incelemiş, ardından önerdiğimiz graf temsilli yöntemle elde edilen deney sonuçları, ilk sonuçlarla birlikte karşılaştırmalı olarak sunulmuştur. Sonuçlar, geleneksel yöntem için BERT kelime gömme modelinin yüksek öznitelik sayıları ile iyi sonuç verdiğini, bununla beraber önerdiğimiz graf dönüşümlü yaklaşımın çalışmada en başarılı sonucu verdiğini ortaya koymuştur. Dolayısıyla doğal dil işleme ve sınıflandırma görevlerine son derece olumlu katkıda bulunan

kelime gömme tekniklerinin yanında, graf dönüşümünün de sınıflandırma performansına iyileştirici etkisi ispatlanmıştır.

Gelecekte graf temsilli öğrenme teknikleri, metin temsil vektörleri yerine kelime birlikteliklerine uygulanarak sınıflandırma performansı üzerindeki etkileri araştırılabilir. Aynı zamanda kelimeler yerine n-gram ya da harf düzeyinde graf yapıları oluşturularak sınıflandırma deneyleri yapılabilir. Öte yandan sınıflandırıcı model olarak CNN veya YSA yerine Graph Convolutional Network (GCN) yapıları kullanılarak model sonuçları izlenebilir. Bu tez çalışması metin sınıflandırma alanında graf temsillerinin iyileştirici potansiyelini ortaya koyması açısından gelecekte bu alanda yapılması muhtemel çeşitli varyasyonlara da ışık tutmaktadır.

KAYNAKLAR

1. AYDOĞAN, M. and KARCI, A., "Kelime Temsil Yöntemleri ile Kelime Benzerliklerinin İncelenmesi", ***Çukurova Üniversitesi Mühendislik-Mimarlık Fakültesi Dergisi***, 34 (June): 181–196 (2019).
2. Wang, J., Wang, Z., Zhang, D., and Yan, J., "Combining Knowledge with Deep Convolutional Neural Networks for Short Text Classification", (2017).
3. Sachan, D. S., Zaheer, M., and Salakhutdinov, R., "Revisiting LSTM Networks for Semi-Supervised Text Classification via Mixed Objective Function", ***Proceedings Of The AAAI Conference On Artificial Intelligence***, 33: 6940–6948 (2019).
4. Alam, M., Bie, Q., Türker, R., and Sack, H., "Entity-Based Short Text Classification Using Convolutional Neural Networks", International Conference on Knowledge Engineering and Knowledge Management, 136–146 (2020).
5. Behjati, M., Moosavi-Dezfooli, S.-M., Baghshah, M. S., and Frossard, P., "Universal Adversarial Attacks on Text Classifiers", (2019).
6. Demir, H. and Ozgur, A., "Improving named entity recognition for morphologically rich languages using word embeddings", (2014).
7. Helmy, A. A., Omar, Y. M. K., and Hodhod, R., "An Innovative Word Encoding Method For Text Classification Using Convolutional Neural Network", (2018).
8. Dharma, E. M., Lumban Gaol, F., Leslie, H., Warnars, H. S., and Soewito, B., "THE ACCURACY COMPARISON AMONG WORD2VEC, GLOVE, AND FASTTEXT TOWARDS CONVOLUTION NEURAL NETWORK (CNN) TEXT CLASSIFICATION", ***Journal Of Theoretical And Applied Information Technology***, 31 (2): (2022).
9. Saleh, H., Alhothali, A., and Moria, K., "Detection of Hate Speech using BERT and Hate Speech Word Embedding with Deep Model", ***Applied Artificial Intelligence***, 37 (1): (2023).

10. Sabbeh, S. F. and Fasihuddin, H. A., "A Comparative Analysis of Word Embedding and Deep Learning for Arabic Sentiment Classification", ***Electronics***, 12 (6): 1425 (2023).
11. Kancharapu, R. and A Ayyagari, S. N., "A comparative study on word embedding techniques for suicide prediction on COVID-19 tweets using deep learning models", ***International Journal Of Information Technology (Singapore)***, (2023).
12. Fachrurrozi, S., Shidik, G., … A. F.-… on A. for, and 2021, undefined, "Increasing Accuracy of Support Vector Machine (SVM) By Applying N-Gram and Chi-Square Feature Selection for Text Classification", ***Ieeexplore.Ieee.Org***, .
13. Institute of Electrical and Electronics Engineers. Beijing Section and Institute of Electrical and Electronics Engineers, "ICSESS 2018 : Proceedings of 2018 IEEE 9th International Conference on Software Engineering and Service Science : November 23-25,2018, China Hall of Science and Technology, Beijing, China", .
14. Meesad, P., Boonrawd, P., and Nuipian, V., "A Chi-Square-Test for Word Importance Differentiation in Text Classification", .
15. Zhang, X., Zhao, J., and Lecun, Y., "Character-level convolutional networks for text classification", ***Advances In Neural Information Processing Systems***, 2015-Janua: 649–657 (2015).
16. Joulin, A., Grave, E., Bojanowski, P., and Mikolov, T., "Bag of tricks for efficient text classification", ***15th Conference Of The European Chapter Of The Association For Computational Linguistics, EACL 2017 - Proceedings Of Conference***, 2: 427–431 (2017).
17. Wang, R., Li, Z., Cao, J., Chen, T., and Wang, L., "Convolutional Recurrent Neural Networks for Text Classification", (2019).
18. Amasyalı, M. F., Balcı, S., Mete, E., and Varlı, E. N., "Türkçe Metinlerin Sınıflandırılmasında Metin Temsil Yöntemlerinin Performans Karşılaştırılması / A Comparison of Text Representation Methods for Turkish Text Classification", ***EmBilimselDergi***, 2 (4): (2012).
19. Rousseau, F. and Kiagias, E., "Text Categorization as a Graph Classification Problem", .

20. Yao, L., Mao, C., and Luo, Y., "Graph Convolutional Networks for Text Classification", .
21. Yanardag, P. and Vishwanathan, S. V. N., "Deep graph kernels", ***Proceedings Of The ACM SIGKDD International Conference On Knowledge Discovery And Data Mining***, 2015-Augus: 1365–1374 (2015).
22. Perozzi, B., Al-Rfou, R., and Skiena, S., "DeepWalk: Online Learning of Social Representations", .
23. Ghadiri, N., Samani, R., and Shahrokh, F., "Integration of Text and Graph-Based Features for Depression Detection Using Visibility Graph", 332–341 (2023).
24. Li, J., Yang, Y., Wu, Z., Vydiswaran, V. G. V., and Xiao, C., "ChatGPT as an Attack Tool: Stealthy Textual Backdoor Attack via Blackbox Generative Model Trigger", (2023).
25. Yogatama, D., Dyer, C., Ling, W., and Blunsom, P., "Generative and Discriminative Text Classification with Recurrent Neural Networks", (2017).
26. Sert, E. R., "Word Context and Token Representations from Paradigmatic Relations and Their Application to Part-of-Speech Induction", (2012).
27. Viola, P., Way, O. M., Jones, M. J., and Snow, D., "Detecting pedestrian using patterns of motion and appearance.", ***International Journal Of Computer Vision.***, 63 (2): 153–161 (2005).
28. Zhang, W., Yoshida, T., and Tang, X., "A comparative study of TF*IDF, LSI and multi-words for text classification", ***Expert Systems With Applications***, 38 (3): 2758–2765 (2011).
29. Jain, S., Vishwakarma, S., and Jain, S. C., "Analysis of Term Weighting schemes in Vector Space model for text classification", ***Journal Of Integrated Science And Technology***, 11 (2): 469–469 (2023).
30. Cuello, C. Y., Caradonna, V. J., José, M., Ucelay, G., and Cagnina, L. C., "On the Importance of Data Representation for the Success of Text Classification", (2022).
31. Kabra, B. and Nagar, C., "Convolutional Neural Network based sentiment analysis with TF-IDF based vectorization", ***Journal Of Integrated Science And Technology***, 11 (3): 503–503 (2023).

32. Madatov, K., Bekchanov, S., and Vičič, J., "Uzbek text summarization based on TF-IDF", (2023).
33. Çelikkol, P., Laubrock, J., and Schlangen, D., "TF-IDF based Scene-Object Relations Correlate With Visual Attention", ***2023 Symposium On Eye Tracking Research And Applications***, 1–6 (2023).
34. Xu, S., Leng, Y., Feng, G., Zhang, C., and Chen, M., "A gene pathway enrichment method based on improved TF-IDF algorithm", ***Biochemistry And Biophysics Reports***, 34: 101421 (2023).
35. Wei, C., Wang, B., and Kuo, C. C. J., "Task-specific dependency-based word embedding methods", ***Pattern Recognition Letters***, 159: 174–180 (2022).
36. Rahimi, Z. and Homayounpour, M. M., "The impact of preprocessing on word embedding quality: a comparative study", ***Language Resources And Evaluation***, 57 (1): 257–291 (2022).
37. Nanni, L., Brahnam, S., Ghidoni, S., Menegatti, E., and Barrier, T., "Different Approaches for Extracting Information from the Co-Occurrence Matrix", ***PLOS ONE***, 8 (12): e83554 (2013).
38. Almeida, F. and Xexéo, G., "Word Embeddings: A Survey", (2019).
39. Mikolov, T., Grave, E., Bojanowski, P., Puhrsch, C., and Joulin, A., "Advances in Pre-Training Distributed Word Representations", ***LREC 2018 - 11th International Conference On Language Resources And Evaluation***, 52–55 (2017).
40. Goldberg, Y., Levy, O., Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean, J., "Word2vec Explained: deriving Mikolov et al.'s negative-sampling word-embedding method", (2014).
41. Hung, P. T. and Yamanishi, K., "Word2vec Skip-Gram Dimensionality Selection via Sequential Normalized Maximum Likelihood", ***Entropy 2021, Vol. 23, Page 997***, 23 (8): 997 (2021).
42. Peng, H., Li, J., Yan, H., Gong, Q., Wang, S., Liu, L., Wang, L., and Ren, X., "Dynamic network embedding via incremental skip-gram with negative sampling", ***Science China Information Sciences***, 63 (10): 1–19 (2020).
43. Preethi Krishna, P. and Sharada, A., "Word Embeddings - Skip Gram Model", ***ICICCT 2019 – System Reliability, Quality Control, Safety, Maintenance And Management***, 133–139 (2020).

44. Cuba Gyllensten, A. and Sahlgren, M., "Shallow Contextualized Word Embeddings", (2022).
45. Andrabi, S. A. B. and Wahid, A., "A Comparative Study of Word Embedding Techniques in Natural Language Processing", 701–712 (2022).
46. Kowsher, M., Sobuj, M. S. I., Shahriar, M. F., Prottasha, N. J., Arefin, M. S., Dhar, P. K., and Koshiba, T., "An Enhanced Neural Word Embedding Model for Transfer Learning", ***Applied Sciences 2022, Vol. 12, Page 2848***, 12 (6): 2848 (2022).
47. Pennington, J., Socher, R., and Manning, C. D., "GloVe : Global Vectors for Word Representation", 1532–1543 (2014).
48. Dharma, E. M., Lumban Gaol, F., Leslie, H., Warnars, H. S., and Soewito, B., "THE ACCURACY COMPARISON AMONG WORD2VEC, GLOVE, AND FASTTEXT TOWARDS CONVOLUTION NEURAL NETWORK (CNN) TEXT CLASSIFICATION", ***Journal Of Theoretical And Applied Information Technology***, 31 (2): (2022).
49. Ranade, A., Telge, S., and Mate, Y., "Predicting Disasters from Tweets Using GloVe Embeddings and BERT Layer Classification", ***Communications In Computer And Information Science***, 1528 CCIS: 492–503 (2022).
50. Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T., "Enriching Word Vectors with Subword Information", ***Transactions Of The Association For Computational Linguistics***, 5: 135–146 (2017).
51. Mikolov, T., Grave, E., Bojanowski, P., Puhrsch, C., and Joulin, A., "Advances in Pre-Training Distributed Word Representations", ***LREC 2018 - 11th International Conference On Language Resources And Evaluation***, 52–55 (2017).
52. Su, Y., Lin, R., and Jay Kuo, C. C., "Tree-structured multi-stage principal component analysis (TMPCA): Theory and applications", ***Expert Systems With Applications***, 118: 355–364 (2019).
53. Kaur, K. and Kaur, P., "Improving BERT model for requirements classification by bidirectional LSTM-CNN deep model", ***Computers And Electrical Engineering***, 108: (2023).
54. Prabhu, S., Mohamed, M., and Misra, H., "Multi-class Text Classification using BERT-based Active Learning", (2021).

55. Devlin, J., Chang, M.-W., Lee, K., Google, K. T., and Language, A. I., "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding", .

56. Onan, A., "Hierarchical graph-based text classification framework with contextual node embedding and BERT-based dynamic fusion", ***Journal Of King Saud University - Computer And Information Sciences***, 35 (7): 101610 (2023).

57. Meesad, P., Boonrawd, P., and Nuipian, V., "A Chi-Square-Test for Word Importance Differentiation in Text Classification", ***International Conference On Information And Electronics Engineering***, 6 (February 2015): 110–114 (2011).

58. Kuang, S. and Davison, B. D., "Learning word embeddings with chi-square weights for healthcare tweet classification", ***Applied Sciences (Switzerland)***, 7 (8): 846 (2017).

59. Türker, İ. and Aksu, S., "Connectogram – A graph-based time dependent representation for sounds", ***Applied Acoustics***, 191: (2022).

60. Albert, R. and Barabási, A. L., "Statistical mechanics of complex networks", ***Reviews Of Modern Physics***, 74 (1): 47 (2002).

61. Mata, A. S. da, "Complex Networks: a Mini-review", ***Brazilian Journal Of Physics***, 50 (5): 658–672 (2020).

62. Albert, R. and Barabasi, A.-L., "Statistical mechanics of complex networks", ***Reviews Of Modern Physics***, 74 (1): 47–97 (2001).

63. Türker, İ., "Generating clustered scale-free networks using Poisson based localization of edges", ***Physica A: Statistical Mechanics And Its Applications***, 497: 72–85 (2018).

64. Çavuşoğlu, A. and Türker, I., "Scientific collaboration network of Turkey", ***Chaos, Solitons And Fractals***, 57: 9–18 (2013).

65. Türker, İ., Şehirli, E., and Demiral, E., "Uncovering the differences in linguistic network dynamics of book and social media texts", ***SpringerPlus***, 5 (1): (2016).

66. Darbaş, H., Karci, A., Tbmyo, A. Ü., and Programcılığı, B., "Graf Benzerliği İle Metin Kiyaslama", (2020).

67. Hu, Y. and Xiao, F., "A novel method for forecasting time series based on directed visibility graph and improved random walk", ***Physica A: Statistical Mechanics And Its Applications***, 594: (2022).
68. Wen, T., Chen, H., and Cheong, K. H., .
69. Zafari, A., Kianmehr, M. H., and Abdolahzadeh, R., "Modeling the effect of extrusion parameters on density of biomass pellet using artificial neural network", ***International Journal Of Recycling Of Organic Waste In Agriculture***, 2 (1): (2013).
70. Nithya, A., Appathurai, A., Venkatadri, N., Ramji, D. R., and Anna Palagan, C., "Kidney disease detection and segmentation using artificial neural network and multi-kernel k-means clustering for ultrasound images", ***Measurement***, 149: 106952 (2020).
71. Goodfellow, I., Bengio, Y., and Courville, A., "Deep Learning [pre-pub version]", (2016).
72. Mostafa, B., El-Attar, N., Abd-Elhafeez, S., and Awad, W., "Machine and Deep Learning Approaches in Genome: Review Article", ***Alfarama Journal Of Basic & Applied Sciences***, 0 (0): 0–0 (2020).
73. Safaya, A., "BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ BSM 401 BİLGİSAYAR MÜHENDİSLİĞİ TASARIMI DERİN ÖĞRENME TABANLI STACK OVERFLOW CHATBOT", .
74. Srivastava, N., Hinton, G., Krizhevsky, A., and Salakhutdinov, R., "Dropout: A Simple Way to Prevent Neural Networks from Overfitting", (2014).
75. ZahediNasab, R. and Mohseni, H., "Neuroevolutionary based convolutional neural network with adaptive activation functions", ***Neurocomputing***, 381: 306–313 (2020).

ÖZGEÇMİŞ

Elif DORUKBAŞI, 75.Yıl Karabük Anadolu Lisesinden 2009 yılında mezun oldu. Aynı yıl Kocaeli Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği bölümüne başlayarak, 2013 yılında iyi bir derece ile mezun oldu. Mezuniyet sonrası özel bir bankada yazılım mühendisi olarak çalışmaya başladı. 2020 yılında Karabük Üniversitesi Lisansüstü Eğitim Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı'nda yüksek lisans eğitimine başlamış olup aynı kurumda araştırma görevlisi olarak çalışmaya devam etmektedir.