İSTANBUL TOPKAPI ÜNİVERSİTESİ | LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

MEVCUT DERİN DOĞAL DİL İŞLEME MODELLERİNİN TÜRKÇE ÖZELİNDE PERFORMANSLARININ ARTIRILMASI

DOKTORA TEZİ

BİLAL KOBANOĞLU

Yönetim Bilişim Sistemleri Anabilim Dalı

Yönetim Bilişim Sistemleri Bilim Dalı

NİSAN, 2025

İSTANBUL TOPKAPI ÜNİVERSİTESİ | LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

MEVCUT DERİN DOĞAL DİL İŞLEME MODELLERİNİN TÜRKÇE ÖZELİNDE PERFORMANSLARININ ARTIRILMASI

DOKTORA TEZİ

BİLAL KOBANOĞLU

21220903002

Yönetim Bilişim Sistemleri Anabilim Dalı

Yönetim Bilişim Sistemleri Bilim Dalı

Tez Danışmanı: Prof. Dr. Fazlı Yıldırım

NİSAN, 2025

22/04/2025

KABUL VE ONAY

Enstitümüz Yönetim Bilişim Sistemleri Anabilim Dalı, Yönetim Bilişim Sistemleri (Doktora) Programı 21220903002 numaralı öğrencisi Bilal KOBANOĞLU'nun doktora tez savunma sınav jürisine ilişkin Yönetim Bilişim Sistemleri Anabilim Dalı Başkanlığının önerisi görüşüldü. Yapılan görüşme sonunda; adı geçen öğrencinin **"MEVCUT DERİN DOĞAL DİL İŞLEME MODELLERİNİN TÜRKÇE ÖZELİNDE PERFORMANSLARININ ARTIRILMASI"** başlıklı Doktora tezi Enstitümüz Yönetim Kurulunun 24/03/2025 tarihli ve 2025/08 sayılı Yönetim Kurulu kararıyla oluşturulan jüri tarafından oybirliği / oyçokluğu ile 22/04/2025 tarihinde kabul edilmiştir.

	Unvan	Adı Soyadı	Üniversite	İmza
		ASIL ÜYELER		
Danışman	Prof. Dr.	Fazlı YILDIRIM	İstanbul Topkapıı Üniversitesi	
1.Üye	Prof. Dr.	Batuhan KOCAOĞLU	İstanbul Topkapıı Üniversitesi	
2.Üye	Dr. Öğr. Üyesi	Güneş KÜÇÜKYAZICI	İstanbul Topkapıı Üniversitesi	
3.Üye	Dr. Öğr. Üyesi	Yasemin DEMİREL	Fenerbahçe Üniversitesi	
4.Üye	Dr. Öğr. Üyesi	Ali KILINÇ	Piri Reis Üniversitesi	
		YEDEK ÜYELER		
1. Üye	Dr. Öğr. Üyesi	Engin HENGİRMEN	İstanbul Topkapı Üniversitesi	
2. Üye	Prof. Dr.	Metin ZONTUL	Sivas Bilim ve Teknoloji Üniversitesi	

ONAY
Prof. Dr. Özlem KUNDAY
Enstitü Müdürü

(*) Oybirliği/Oyçokluğu hâli yazı ile yazılacaktır.
(**) Kabul / Ret veya Düzeltme kararı hâli yazı ile yazılacaktır.

AKADEMİK DÜRÜSTLÜK BEYANI

Doktora Tezi olarak sunduğum “Mevcut Derin Doğal Dil İşleme Modellerinin Türkçe Özelinde Performanslarının Artırılması” başlıklı çalışmanın, bilimsel ahlak ve geleneklere uygun olarak tarafımdan yazıldığını, yararlandığım eserlerin tamamının kaynaklarda gösterildiğini ve çalışmam içerisinde kullanıldıkları her yerde atıf yapıldığını belirtir ve onurumla doğrularım.

22 Nisan 2025
Bilal Kobanoğlu

TEŞEKKÜR

Yerleri, gökleri ve arasındakileri kusursuz bir düzende yaratan ve onları zaman içinde sonsuz hikmet ve kudretiyle belirli bir amaç üzere yörüngelerinde tutan, insanoğluna sürekli akletmeyi, düşünmeyi ve dosdoğru olmayı tavsiye ederek bizlere evrenin işleyişine dair derin bir farkındalık kazandıran Allah'a şükürlerimi sunarım.

Doktora sürecimde bana sürekli destek olan, çözüm ve kolaylık odaklı anlayışıyla etrafına sürekli pozitif enerji saçarak beni motive eden tez danışmanım ve bölüm başkanımız Prof. Dr. Fazlı Yıldırım hocama teşekkürlerimi sunarım.

Yine bu süreçte bana destek olan eşim Ayşegül'e ve babalarıyla geçirmeleri gereken süreden feragat ederek bana çalışmalarım için vakit tanıyan oğullarım Selim Agâh ve Yusuf Aras'a müteşekkirim.

Hayatımın her aşamasında olduğu gibi doktora sürecinde de manevi desteklerini hep arkamda hissettiğim annem Fatma Kobanoğlu'na ve babam Behçet Kobanoğlu'na minnettarlığımı buradan bir kere daha ifade etmek isterim.

İÇİNDEKİLER

ŞEKİLLER LİSTESİ

KISALTMALAR

AI	:	Artificial Intelligence
BERT	:	Bidirectional Encoder Representation from Transformers
BoW	:	Bag of Words
BPE	:	Byte-Pair Encoding
CBOW	:	Continuous Bag of Words
CNN	:	Convolutional Neural Networks
CTC	:	Connectionist Temporal Classification
ELMo	:	Embeddings from Language Models
ERNIE	:	Enhenced Representation through Knowledge Integration
GPT	:	Generative Pretrained Transformer
GPU	:	Graphics Processing Unit
GRU	:	Gated Recurrent Unit
IoT	:	Internet of Things
KNN	:	K-Nearest Neighbor
LCS	:	Longest Common Subsequence
LDA	:	Latent Dirichlet Allocation
LSI	:	Latent Semantic Index
LSTM	:	Long Short-Term Memory
NC	:	News Classification
NER	:	Named Entity Recognition
NLG	:	Natural Language Understanding
NLI	:	Natural Language Inference
NLP	:	Natural Language Processing
NLTK	:	Natural Language Toolkit
NLU	:	K-Nearest Neighbor
OpenAI GPT	:	Open Artificial Intelligence Generative Pre-trained Transformer
PLSI	:	Probabilistic Latent Semantic Index
PoS	:	Part of Speech
QA	:	Question Answering
RNN	:	Recurrent Neural Network
ROUGE	:	Recall-Oriented Understudy for Gisting Evaluation
ROUGE-L	:	Recall-Oriented Understudy for Gisting Evaluation-Longest Common Subsequence
SA	:	Sentiment Analysis
seq2seq	:	Sequence-to-seque
SP	:	Syntactic Parsing
ST	:	SentenceTransformer
SVM	:	Support Vector Machine
T5	:	Google Pre-Trained T5 Large Büyük Dil Modeli

TC	:	Text Classification
TF-IDF	:	Term Frequency-Inverse Document Frequency
TL	:	Topic Labelling
ULM	:	Unigram language model
YZ	:	Yapay Zekâ

ÖZET

MEVCUT DERİN DOĞAL DİL İŞLEME MODELLERİNİN TÜRKÇE ÖZELİNDE PERFORMANSLARININ ARTIRILMASI

Bu tez, Türkçe metin özetleme alanında mevcut derin doğal dil işleme modeli olan T5 modelinin metin içindeki cümlelerin kosinüs benzerliklerini kullanarak özetleme performansını artırmayı amaçlamaktadır. Türkçe metinlerin doğru ve etkili bir şekilde özetlenmesi, doğal dil işleme (NLP) alanında önemli bir zorluk oluşturmakta olup, bu çalışma, özetleme süreçlerinde model performansını iyileştirmek için yeni yaklaşımlar sunmaktadır. Çalışmanın temel hedefi, özetlenecek metnin cümleler arasındaki kosinüs benzerliklerini hesaplayarak, önemli cümlelerin seçilmesi ve bunların T5-large modeline entegre edilmesiyle, özetleme sonuçlarının kalitesinin nasıl iyileştirilebileceğini incelemektir.

Bu araştırmanın kuramsal dayanağı dilin dağılımsal düzenliliklerinin sayısal temsillerle ifadesini kullanarak, bu düzenliliklerin benzerlik oranına göre üretken dil modellerinin daha düzenli çıktı üretmesi üzerinedir. Bu bağlamda her kelimenin bir vektör olarak ifade edildiği uzayda, cümlelerin oluşturduğu bağlamsal vektörler arasındaki benzerlik (düzenlilik) kullanılarak, büyük dil modellerinin daha düzenli üretken çıktı üretmesi beklenmektedir. Bu diktonomi yaklaşımı, giriş vektörler arasındaki benzerliklerin başka bir süreçte hesaplanarak giriş vektöründeki düzenliliklerin artırımı ile üretken çıktının daha düzenli olmasını hedeflemektedir.

Araştırmada, diktonomi yaklaşımı benimsenmiş ve bu yaklaşımın etkisiyle metinler, cümlelerin önem derecelerine göre sıralanarak özetleme işlemi gerçekleştirilmiştir. Modelin başarısı, ROUGE metrikleri kullanılarak değerlendirilmiş ve elde edilen sonuçlar üzerinden, kosinüs benzerliklerinin T5-Large ile birlikte kullanıldığında, özetleme performansında anlamlı bir iyileşme sağladığı tespit edilmiştir.

Bu çalışmanın bulguları, Türkçe metin özetleme alanında kullanılabilecek yenilikçi bir yaklaşım sunarak hem akademik literatüre katkı sağlamakta hem de pratik uygulamalarda daha etkili özetleme tekniklerinin geliştirilmesine olanak tanımaktadır.

Anahtar Kelimeler: Yapay Zekâ, Derin Öğrenme, Türkçe Metin Özetleme, Önceden Eğitilmiş Dil Modelleri, Vektörlerin Kosinüs Benzerlikleri, T5 Model, Doğal Dil İşleme, ROUGE Skoru

ABSTRACT

IMPROVING THE PERFORMANCE OF EXISTING DEEP NATURAL LANGUAGE PROCESSING MODELS IN THE CONTEXT OF TURKISH

This thesis aims to enhance the summarization performance of the T5 model, a state-of-the-art deep natural language processing model, by utilizing the cosine similarity between sentences within a text for Turkish text summarization. Accurately and effectively summarizing Turkish texts presents a significant challenge in the field of natural language processing (NLP), and this study offers new approaches to improve model performance in summarization tasks. The primary objective of this research is to examine how summarization results can be improved by calculating the cosine similarities between sentences in the text to select important sentences and integrate them into the T5-large model.

The theoretical foundation of this research is based on the expression of the distributional regularities of language through numerical representations, aiming to improve the output of generative language models by using the similarity ratio of these regularities. In this context, by representing each word as a vector in a space, and using the similarity (regularity) between contextual vectors formed by sentences, it is expected that large language models will produce more regular generative outputs. This diktonomy approach aims to enhance the regularity of the generative output by calculating the similarities between input vectors in another process, thereby increasing the regularities in the input vectors.

In this study, the dichotomy approach is adopted, where texts are ranked based on the importance of their sentences, and summarization is performed accordingly. The model's performance is evaluated using ROUGE metrics, and the results indicate that using cosine similarities alongside T5-large leads to a significant improvement in summarization performance.

The findings of this study provide an innovative approach for Turkish text summarization, contributing to the academic literature and enabling the development of more effective summarization techniques for practical application

Keywords: Artificial Intelligence, Deep Learning, Turkish Text Summarization, Pretrained Language Models, Cosine Similarities of Vectors, T5 Model, Natural Language Processing, ROUGE Score

GİRİŞ

Bu araştırmanın genel amacı, mühendislik prensiplerini ve ayrık düşünme yöntemlerini kullanarak, mevcut araçlarla problemlere daha verimli ve etkili çözümler bulmak ve aynı zamanda Yeşil Mutabakata (Avrupa Komisyonu, 2019) uyum sağlamaktır.

Türkçe metin özetleme, doğal dil işleme (NLP) alanında önemli bir araştırma konusu olup, metinlerin kısa, öz ve anlamlı bir şekilde özetlenmesi işlemidir. Ancak, Türkçe'nin yapısal farklılıkları ve dil özellikleri, özetleme süreçlerini daha karmaşık hale getirebilmektedir. Bu nedenle, Türkçe metin özetleme görevlerinde daha etkili sonuçlar elde edebilmek için, son yıllarda yapay zekâ ve derin öğrenme yöntemleri üzerine yoğunlaşan araştırmalar artmıştır. Özellikle, önceden eğitilmiş dil modelleri metinlerin daha doğru özetlenmesi için önemli araçlar haline gelmiştir.

Bu araştırmanın geleneksel yöntemlere farklı bakış açısı kazandırarak, önceden eğitilmiş T5 büyük dil modelinin performansını, Türkçe metin özetleme sırasında özetlenecek metin içindeki cümleler arasındaki vektörel kosinüs benzerlikleri en yüksek olan cümlelerin seçilerek modelin performansının iyileştirilmesidir. Bu yaklaşım, modelin daha anlamlı ve doğru özetler üretmesini sağlamayı hedeflemektedir. Kosinüs benzerliği, cümlelerin anlamını karşılaştıran bir teknik olup, seçilen cümlelerin metnin genel anlamını daha iyi yansıtmaya yardımcı olabilir.

Bu araştırmanın temel amacı, Türkçe metinlerin, var olan büyük dil modelleriyle özetleme alanındaki sınırlı literatüre katkı sağlamayı, aynı zamanda büyük dil modellerinin Türkçe 'yi daha verimli bir şekilde işleyebilmesini hedefleyen yenilikçi bir çözüm önerisi sunmayı amaçlamaktadır.

Literatürde, büyük dil modellerinin Türkçe verileriyle eğitilmesi ve ince ayarlanması üzerine pek çok çalışma bulunmaktadır. Bu çalışmalar, Türkçe'nin kendine has dil bilgisel ve yapısal özelliklerinin dikkate alınarak büyük dil modellerinin

performansının iyileştirilmesine yönelik çeşitli metodolojiler geliştirmiştir. Türkçe metinlere özgü dil özelliklerinin doğru bir şekilde modellenebilmesi için, önceden eğitilmiş büyük dil modellerinin Türkçe veri setleriyle ince ayar yapılarak, dilin zengin yapısının daha etkin bir şekilde öğrenilmesi sağlanmıştır. Bu bağlamda, dilin morfolojik, sentaktik ve semantik özelliklerini doğru bir şekilde yakalamayı hedefleyen metotlar, modelin Türkçe verileriyle uyumlu hale gelmesini sağlamıştır.

Türkçe dilindeki performansı artırmaya yönelik diğer bir yaklaşım ise, modelin Türkçe veri setlerinde daha iyi genelleme yapabilmesi için ince ayar sırasında kullanılan tekniklerde yapılan özelleştirmelerdir. Özellikle, büyük dil modellerinin Türkçe verilerine adapte edilmesi sürecinde yeni model eğitme, transfer öğrenme, veri artırma ve dil özelliklerine duyarlı parametre optimizasyonu gibi stratejiler, modelin doğruluğunu ve verimliliğini önemli ölçüde iyileştirmiştir. Bu çalışmalar, büyük dil modellerinin Türkçe metin işleme alanındaki performansını artırmaya yönelik somut çözümler sunmakta ve daha güçlü, Türkçe'ye özgü metin işleme sistemlerinin geliştirilmesine olanak tanımaktadır.

Kartal ve Kutlu (2020) kendi modellerini eğitirken cümle seçiminde, Bakan ve Yakut (2024) graf tabanlı özetlemede, Karakoç ve Yılmaz (2019) kendi modellerini eğitirken kosinüs benzerlik tabanlı içerik seçimi yapmaktadırlar. Şakar ve Emekci (2025) farklı veri kümelerinde, vektör veri tabanlarında, RAG yöntemlerinde, büyük dil modellerinde (LLM'ler), gömme modelleri ve gömme filtre puanları üzerinde bir grid search optimizasyonu üzerine çalışmış ve RAG performansı, her bir veri kümesi ve soru-cevap çiftinden elde edilen gömülü LLM yanıtı ile referans yanıtı arasındaki fark kosinüs benzerliğini ölçerek değerlendirmiştir. Gopalakrishnan vd. (2025) GPT-4 modelinin tıbbi yönergelerden neden-sonuç ilişkilerinin çıkarılması üzerine yaptıkları çalışmada modelin doğruluğunu değerlendirmek için kosinüs benzerliği gibi alternatif yöntem kullanmıştır. Bu çalışmalar, literatürdeki kendi modelleri ile Türkçe özelinde metin özetleme üzerine olan sınırlı sayıdaki çalışmalardır.

Literatürde yer alan büyük dil modellerinin Türkçe verileriyle eğitilmesi ve ince ayarlanması üzerine yapılan çalışmalar genellikle yüksek işlem gücü gereksinimi ve bu süreçlerin çevresel etkileri açısından bazı sınırlamalara sahiptir. Bu modellerin eğitilmesi ve optimize edilmesi, büyük miktarda verinin işlenmesi ve model parametrelerinin ayarlanması sürecinde önemli bilgi işlem gücü ve hesaplama kaynakları talep etmektedir. Bu durum, özellikle derin öğrenme tabanlı modellerin büyük veri setlerinde eğitim süreçlerinde hesaplama sürelerinin uzamasına ve bu süreçlerde büyük miktarda enerji tüketilmesine yol açmaktadır.

Ayrıca, bu yüksek işlem gücü gereksinimi, çevresel etkiler ve karbon salınımı açısından da önemli bir problem teşkil etmektedir. Büyük dil modellerinin eğitilmesi, büyük veri merkezlerinde yoğun hesaplama kaynaklarının kullanılmasına ve bu süreçlerin sonucunda ciddi bir enerji tüketimi ve karbon salımı meydana gelmesine neden olmaktadır.

1 YAPAY ZEKÂ

1.1 Yapay Zekâ Kavramı

Yapay zekâ, beşeri düşünce yeteneklerini bilgisayarlara kazandırarak insanların yaptığı işleri bilgisayarların yapabilecek hale gelmesini sağlayan bir bilim dalıdır. Aslında yapay zekâ bilgisayar bilimlerinin, insanların genellikle zorlu, karmaşık ve rutin olarak kabul edilen görevleri, bilgisayarlar aracılığıyla gerçekleştirilebilecek hale getirme çabasıdır (Russell ve Norvig, 2010).

Yapay zekâ, tipik olarak insan zekâsı gerektiren görevleri yerine getirebilen akıllı makineler yaratmayı amaçlayan bir bilgisayar bilimi dalıdır. Bu görevler arasında doğal dili anlama, kalıpları tanıma, deneyimlerden öğrenme ve karar verme yer alır (Wahl vd., 2018).

Yapay zekâ, makine öğrenimi, sinir ağları, doğal dil işleme ve robotik gibi birçok teknoloji ve yaklaşımı içeren geniş bir yelpazeyi kapsar. Bu teknolojiler, makinelerin karmaşık veriyi analiz etmelerini, konuşma ve görüntüleri tanımalarını ve hatta insan karar verme süreçlerini taklit etmelerini sağlar (Khakurel vd., 2018).

Yapay zekâ, bir sistemin dış verileri yorumlama, bu verilerden öğrenme ve esnek adaptasyon yoluyla belirli hedeflere ve görevlere ulaşma yeteneğini ifade etmektedir (Feng vd., 2021).

Yapay zekâ, tüm endüstrilerin verimlilik kazanımlarından ve maliyet düşüşlerinden yararlanmak için kullanılmak istenen, hızla büyüyen bir teknolojik olgudur. Yapay zekâ, otomasyon, veri analizi ve öğrenme yetenekleri gibi özellikleriyle iş süreçlerini optimize etme ve daha etkili kararlar alma potansiyeli sunar. Bu, çeşitli sektörlerde iş süreçlerinin daha verimli hale getirilmesini, kaynakların daha etkili kullanılmasını ve genel olarak rekabet avantajı elde edilmesini sağlar (Hassani vd., 2020).

Yapay zekâ, bir kişinin entelektüel ve yaratıcı işlevlerini yerine getiren, bağımsız olarak sorunları çözmenin yollarını bulan ve karar veren makinelerin, bilgisayar

programlarının ve sistemlerin tamamı olarak kabul edilmektedir (Shabbir ve Anwer, 2018).

Yapay zekâ, ilk kez 1956 yılında bir konferansta ele alınmıştır. Bu konferans, yapay zekânın temellerini atan bilim insanları John McCarthy, Marvin Minsky, Nathaniel Rochester ve Claude Shannon tarafından düzenlenmiştir (Russell ve Norvig, 2010). O tarihten bu yana, yapay zekâ alanında birçok gelişme yaşanmıştır. Özellikle 2010'lardan itibaren, derin öğrenme gibi yeni teknolojilerin kullanımı sayesinde yapay zekâ uygulamaları daha da yaygın hale gelmiştir (Goodfellow vd., 2016). Bugün, yapay zekâ teknolojileri birçok alanda kullanılmaktadır. Örneğin, sağlık sektöründe yapay zekâ, tanı ve tedavi süreçlerinde kullanılmaktadır. Yapay zekâ algoritmaları, hastalıkların tanısında ve tedavisinde daha doğru sonuçlar elde edilmesini sağlamaktadır (Topol, 2019).

Yapay zekâ, Nesnelerin İnternet'ine (IoT) eklendiğinde cihazların insan müdahalesi olmadan verileri analiz etme, karar verme ve bu veriler üzerinde hareket etme yeteneği de dâhil olmak üzere çeşitli yönleri kapsar (Nozari vd., 2022). Aynı zamanda insan beyninin akıl yürütme, tanımlama, anlama, katılım, öğrenme, düşünme ve problem çözme faaliyetlerini taklit etmek için bilgisayarların nasıl kullanılacağına ilişkin çalışmayı da içerir (Shan ve Yu, 2021). Ayrıca yapay zekâ, doğal dili akıllı işlemeyi ve profesyonel yöntemleri içeren bir bilgisayar bilimi dalıdır (Wang ve Fu, 2021).

Özetle yapay zekâ, sistemlerin verileri yorumlama, öğrenme ve belirli hedeflere ulaşma, insan zekâsını taklit etme ve entelektüel ve üretici işlevleri bağımsız olarak gerçekleştirme yeteneğini kapsamaktadır. Çeşitli sektörlerdeki çeşitli uygulamalarıyla hızla büyüyen bir teknolojik olgudur.

1.1 Makine Öğrenimi

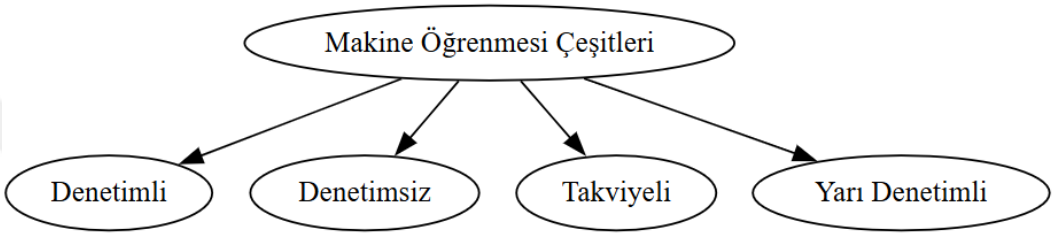
Makine öğrenimi, bilgisayarların belirli bir görevde performanslarını deneyim yoluyla ve açık programlama olmaksızın geliştirmelerine odaklanan yapay zekâ (AI) alanının bir alt kümesidir (Htun ve Tun, 2018). Makine öğrenimi, yapay zekâ

araştırmasının temel konusu olarak kabul edilmekte olup istenmeyen e-posta filtreleri, web arama, kredi puanlaması, sahtekârlık tespiti, fiyat tahmini, reklam yerleştirme, ilaç tasarımı, sağlık, ulaşım gibi çeşitli uygulamalarda kullanılan bir yöntem olarak öne çıkmaktadır (Nassif vd., 2019). Makine öğreniminin amacı, genel amaçlı bilgisayar makinelerinde çalıştırılabilen öğrenme algoritmalarını tasarlamaktır ve bu, insan beyninden ilham almaktır, başka bir deyişle makine öğreniminin amacı, insan beyninden esinlenerek genel amaçlı hesaplamalı makinelerde yürütülebilecek öğrenme algoritmaları tasarlamaktır (Faradonbe vd., 2020).

Makine öğrenimi, makinelere veri sağlama ve onların özerk bir şekilde öğrenmelerine olanak sağlama ve böylece akıllı davranışlar sergileme fikrine dayanmaktadır. Makine öğrenimi, çeşitli alanlarda yaygın olarak kullanılan yapay zekâ araştırmasının temel bir yönüdür. Sistemlerin deneyim yoluyla performanslarını artırmalarını sağlayan algoritmaların ve modellerin geliştirilmesini içerir ve sosyal ağlar, flow sitometrisi (hücre analizi) gibi alanlardaki süreçleri etkili biçimde geliştirmek için potansiyele sahiptir (Nabi ve Xu, 2021).

Veriye dayalı tahminler ve kararlar vermek için bilinen girdilerden matematiksel modeller oluşturmaya yönelik disiplinler arası bir yaklaşım olan makine öğrenimi, çeşitli alanlarda yaygın uygulamalarla dönüştürücü bir kavram haline gelmiştir. Sistem girdileri ve çıktıları arasındaki bilinmeyen bağımlılığı mevcut verilerden tahmin eden bir algoritmadır. Reklam yerleştirme, kredi puanlama, dolandırıcılık tespiti, hisse senedi alım-satımı, ilaç tasarımı, doğal dil işleme, görüntü tanıma, uzman sistemler, taklit (simülasyon), üretim ve diğer birçok uygulama alanı mevcuttur. Makine öğrenimi kavramı, özellikle üretim mühendisliği bağlamında, hem insan hem de makine öğrenimi perspektiflerinden bilgi temsili fikriyle iç içe geçmiş durumdadır. Ayrıca makine öğrenimi, biyolojik, biyomedikal ve davranış bilimlerinde ve siber güvenlik alanlarında umut verici bir teknoloji olarak giderek daha fazla tanınmaktadır (Zhang vd., 2020).

Makine öğrenmesi Şekil 1.1 de gösterildiği şekilde temel olarak dörde ayrılmaktadır (Sarker, 2021).



Şekil 1.1: Makine öğrenmesi çeşitleri

Kaynak: Sarker, 2021

Denetimli öğrenme, makine öğrenmesinde temel bir paradigmadır. Bu yaklaşım, algoritmanın tahminler veya kararlar yapabilmesi için insanlar tarafından oluşturulan etiketli eğitim verilerinden öğrenir. Bu yaklaşım, algoritmanın etiketli verilerden öğrenmesine izin veren insanlar tarafından oluşturulan etiketlere ihtiyaç duyar. Denetimli öğrenme algoritmalarını uygulamak ve uygulamak için standart bir çerçeve sağlayan scikit-learn gibi birkaç son derece popüler yazılım kütüphaneleri bulunmaktadır (Engelen ve Hoos, 2019).

Denetimsiz öğrenme, makine öğrenmesindeki diğer temel bir paradigmalardan biridir. Algoritma, etiketlenmemiş verilerden öğrenerek gizli desenleri veya içsel yapıları keşfeder. Bu yaklaşım, işaretlenmiş etiketlere dayanmaz ve girdi verilerinden anlamlı iç görüler çıkarmayı amaçlar. Birçok çalışma, denetimsiz öğrenmenin çeşitli alanlarda önemini ve çok yönlülüğünü vurgulamıştır (Abdel-Basset vd. 2022). Denetimsiz öğrenme, insan müdahalesine ihtiyaç duymadan etiketlenmemiş veri kümelerini analiz eden bir süreçtir, yani veri odaklı bir yaklaşımdır (Han ve Kamber, 2011). Bu, üretken

özelliklerin çıkarılması, anlamlı eğilim ve yapıların belirlenmesi, sonuçlarda gruplamalar ve keşif amaçları için yaygın olarak kullanılır. En yaygın denetimsiz öğrenme görevleri arasında kümeleme, yoğunluk tahmini, özellik öğrenme, boyut azaltma, ilişki kurallarını bulma, anormallik tespiti bulunmaktadır (Sarker, 2021).

Yarı denetimli öğrenme, yukarıda bahsedilen denetimli ve denetimsiz yöntemlerin bir birleştirilmesi olarak tanımlanabilir, çünkü hem etiketli hem de etiketlenmemiş veriler üzerinde çalışır. Bu nedenle, "denetimsiz öğrenme" ile "denetimli öğrenme" arasında bir yöntemdir. Gerçek dünyada, etiketli veriler birçok bağlamda nadir bulunabilir ve etiketlenmemiş veriler oldukça fazladır; bu durumlarda yarı denetimli öğrenme faydalıdır. Yarı denetimli bir öğrenme modelinin nihai amacı, yalnızca modeldeki etiketli veriler kullanılarak elde edilenden daha iyi bir tahmin sonucu sağlamaktır. Yarı denetimli öğrenmenin kullanıldığı bazı uygulama alanları arasında makine çevirisi, dolandırıcılık tespiti, veri etiketleme ve metin sınıflandırma yer alır (Sarker, 2021).

Takviyeli öğrenme, yazılım ajanları ve makinelerin, etkinliğini artırmak amacıyla belirli bir bağlam veya ortamda en uygun davranışı otomatik olarak değerlendirmesini sağlayan bir tür makine öğrenme algoritmasıdır, yani bir ortam odaklı bir yaklaşımdır (Kaelbling vd., 1996). Bu tür bir öğrenme, ödül veya ceza temellidir ve nihai hedefi, çevresel etkinliklerden elde edilen iç görüler kullanılarak ödülü artırmak veya riski minimize etmek için eylemler almak olarak belirlenmiştir (Mohammed vd., 2016). Takviyeli öğrenme, robotik, otonom sürüş görevleri, üretim ve tedarik zinciri lojistiği gibi karmaşık sistemlerin operasyonel verimliliğini artırmaya veya otomasyonu artırmaya yardımcı olabilen yapay zekâ modellerini eğitmek için güçlü bir araçtır; ancak temel veya doğrudan problemleri çözmek için tercih edilmemektedir (Sarker, 2021).

Özetle Makine öğrenimi, yapay zekâ alanının bir parçası olup, bilgisayarların deneyimlerinden öğrenerek belirli görevlerde performanslarını geliştirmeye odaklanan bir yöntemdir. Temel hedef, genel amaçlı bilgisayar makinelerinde çalışabilen öğrenme algoritmalarını tasarlamaktır. Makine öğrenimi, verilere dayalı tahminler ve kararlar

almak için girdilerden matematiksel modeller oluşturan disiplinler arası bir yaklaşımdır. Bütün endüstrilerde kullanım alanı olduğundan, giderek rağbet gören bir kavram olmuştur.

1.2 Derin Öğrenme

Kökleri yapay sinir ağlarına dayanan bir teknik olan derin öğrenme, yapay zekânın geleceğini yeniden şekillendirmek için güçlü bir araç olarak ortaya çıkmıştır (Ravì vd., 2017). Sağlık hizmetleri (Belhaouari ve İslam, 2021), tıbbi görüntü analizi (Shen vd., 2017) ve bilgisayarlı görme görevleri (Wang vd., 2021) gibi çeşitli alanlarda başarılı sonuçlar göstermiştir. Derin öğrenme, konu sınıflandırması, duygu analizi, soru cevaplama ve dil çevirisi dâhil olmak üzere doğal dil anlamadaki başarısı dikkat çekmektedir (LeCun vd., 2015). Başarısı, minimum düzeyde elle mühendislik gerektirme kabiliyetine atfedilebilir, bu da derin öğrenmeyi artan hesaplama ve veri kullanılabilirliğine uyarlanabilir hale getirir (LeCun vd., 2015). Ayrıca derin öğrenme, insansız hava aracı görüntü analizi (Han vd., 2021), işbirlikçi araç-altyapı sistemlerinde video uygulamaları (Su vd., 2022) ve hatta eğitim (Tian vd., 2022) gibi alanlara da uygulanmıştır. Derin öğrenmenin gücü, büyük verileri, özellik gösterimini ve örüntü tanımayı yönetme yeteneğinde yatmaktadır ve bu da onu makine öğreniminde merkezi bir teknik haline getirmektedir (Yu vd., 2019).

Derin öğrenme, önemli miktarda veriye dayalı bir modelin eğitilmesini ve ardından yeni verilerin sınıflandırılmasını içeren bir makine öğrenimi biçimidir (Han vd., 2021). Çok katmanlı bir algılayıcı yapay sinir ağı algoritmasıdır (Ding vd., 2015). Derin öğrenme kavramı ilk olarak 1976 yılında eğitim alanında ortaya atılmış ve çeşitli alanlardaki önemi vurgulanmıştır (Tian vd., 2022). Derin öğrenme, görüntü işleme, bilgisayarlı görme, doğal dil işleme ve sinyal işlemeye uygulanarak çok yönlülüğü ve geniş kapsamlı uygulamaları ortaya konmuştur (Saufi vd., 2018). Derin öğrenmenin tanıtılması, makine öğrenimini önemli ölçüde etkileyerek tıbbi görüntüleme analizinde

(Debelee vd., 2020), karaciğer sınıflandırılmasında (Napte ve Mahajan, 2022) ve makine dil çevirisinde (Upadhyay, 2017) güçlü uygulamalara yol açmıştır.

Derin öğrenme, doğal dil anlayışını önemli ölçüde etkileyerek konu sınıflandırması, duygu analizi, soru yanıtlama ve dil çevirisi gibi çeşitli görevlerde dikkate değer ilerlemeler göstermiştir (LeCun vd., 2015). Doğal dil işlemede derin öğrenme algoritmalarının etkinliği, son yıllarda BERT ve LSMT gibi uygulamaların metin anlamadaki potansiyellerini ortaya koymasıyla açıkça ortaya çıkmıştır (Wang vd., 2022). Doğal dil anlamada derin öğrenmenin başarısı, kelimeleri ve belgeleri bir vektör uzayında temsil etme yeteneğinin yanı sıra tekrarlayan sinir ağları ve diğer gelişmiş yapılardan faydalanmasına bağlanabilir (Sugiyama, 2019).

Doğal dil anlama alanında derin öğrenmenin uygulanması aynı zamanda yeni model dil anlama tasarımlarının ve yöntemlerinin ortaya çıkmasıyla ilişkilendirilmiş ve doğal dil işleme alanında önemli ilerlemelere yol açmaktadır (Young vd., 2018). Önemli miktarda etiketlenmemiş veri içeren büyük ağlarda dil modellerinin önceden eğitilmesi ve aşağı yönlü görevlerde ince ayar yapılması, doğal dil anlama görevlerinde OpenAI GPT ve BERT tarafından piyasaya sunulan atılımlara olarak sağlamıştır (Sun vd., 2019). Derin öğrenme algoritmalarının literatürdeki bu görüntüsü, doğal dil işlemede pratik başarılar elde ederek bu alandaki baskın paradigmasını daha da sağlamlaştırmaktadır. (Zhou ve Feng, 2017).

Makine öğrenimi ile derin öğrenme arasındaki ilişki, yapay zekâ alanındaki daha geniş manzarayı anlamanın temelini oluşturmaktadır. Makine öğrenimi, multidisipliner bir bilim dalı olarak, mevcut bilgi yapılarını insan öğrenimini taklit ederek geliştirmeye odaklanmaktadır (Arqawi vd., 2022). Naïve Bayes, Destek Vektör Makineleri (SVM), k-En Yakın Komşular (KNN), karar ağaçları gibi çeşitli algoritma kategorilerini içerir (Arqawi vd., 2022). Öte yandan derin öğrenme, çok katmanlı sinir ağlarına dayanan yeni bir öğrenme algoritması olarak ortaya çıkmış ve makine öğrenimi içinde ayrı bir alan olarak kabul edilmektedir (Kuang vd., 2014). Derin öğrenme genellikle makine

öğreniminin daha karmaşık ve yoğun bir formu olarak düşünülür, benzer prensiplerle çalışsa da verinin karmaşık temsilini öğrenmeye odaklanmaktadır (Khalil vd., 2021).

Derin öğrenme, bilgisayarlı görüş, konuşma tanıma ve robotik gibi birçok uygulamada geleneksel makine öğrenimi yöntemlerini geride bırakarak geniş ölçüde dikkat çekmiştir. Derin öğrenmenin başarısı, karmaşık fonksiyonları etkili bir şekilde öğrenme yeteneğine dayanmaktadır ve destek vektör makineleri, rastgele ormanlar ve tek gizli katmanlı sinir ağları gibi geleneksel makine öğrenimi tekniklerinin sınırlamalarını aşmaktadır. Ayrıca, derin öğrenme yeni model tasarımları ve yöntemleri tanıtarak, özellikle doğal dil işleme ve görüntü tanıma gibi çeşitli alanlarda önemli ilerlemelere öncülük etmiştir (Schmidhuber, 2015).

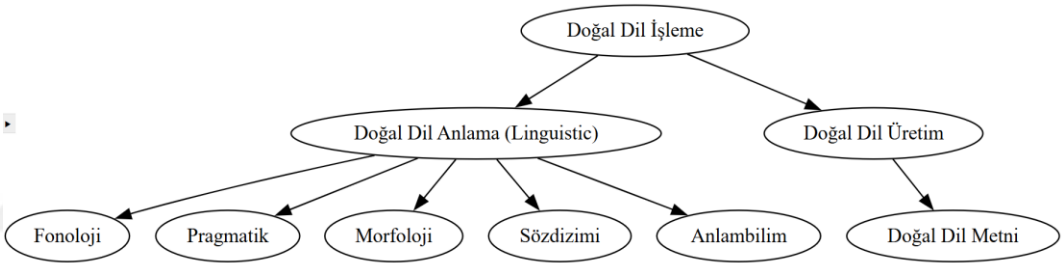
Makine öğrenimi, mühendislik ve deneysel bilimlerde bir standart haline gelirken, derin öğrenme yapay zekâ yeteneklerini özellikle karmaşık görevlerin ve büyük ölçekli veri analizinin üstesinden gelme konusunda devrim yaratmıştır. Makine öğrenimi ile derin öğrenme arasındaki ilişki, sürekli gelişmelerin bir devamı olarak nitelendirilebilir. Derin öğrenme, makine öğreniminin daha karmaşık ve detaylı bir alt kümesini temsil eder ve geleneksel makine öğrenimi yöntemlerinin zorlandığı karmaşık sorunları çözebilme yeteneğine sahiptir (Berral-Garcia, 2018).

Sonuç olarak derin öğrenme, sağlık hizmetlerinden eğitime kadar çeşitli alanlarda köklü değişikliklere neden olmuştur. Büyük verileri işleme, özellik temsili ve örüntü tanıma yeteneği nedeniyle makine öğreniminde merkezi bir teknik haline geldi. Yapay zekâ üzerindeki etkisi ve geniş kapsamlı uygulamaları onu önemli bir araştırma ve geliştirme alanı haline getirerek gelecek için yatırım yapılabilir bir alan olarak görülmesini sağladı.

1.3 Doğal Dil İşleme

Doğal Dil İşleme (Natural Language Programming, NLP), bilgisayarların insan dillerinde yazılmış ifadeleri veya kelimeleri anlamalarını sağlamaya odaklanan Yapay

Zekâ ve Dilbilim alanında bir disiplindir. Bu, kullanıcının işini kolaylaştırmak ve bilgisayarla doğal dilde iletişim kurma isteğini karşılamak amacıyla ortaya çıkmakta ve Doğal Dil Anlama veya Dilbilim ile Doğal Dil Üretimi olmak üzere iki ana kategoriye ayrılmaktadır. Doğal Dil İşleme (NLP), Şekil 1.2 de görüleceği üzere iki temel bileşeni içermektedir: Doğal Dil Anlama (NLU) veya Dilbilim ve Doğal Dil Üretimi (NLG). Bu bileşenler, bilgisayarların insan dilini anlamasını ve üretmesini sağlamada önemli roller oynamaktadır (Khurana vd., 2023).



Şekil 1.2: Doğal Dil İşleme Bileşenleri

Kaynak: Khurna vd., 2023

1.3.1 Doğal Dil Anlama

Doğal Dil Anlama, makinelerin doğal dili anlamasını ve kavramları, varlıkları, duyguları, anahtar kelimeleri çıkararak analiz etmelerini sağlamaktadır. Müşteri hizmetleri uygulamalarında, müşteriler tarafından sözlü veya yazılı olarak bildirilen sorunları anlamak için kullanılır. Dilbilim, dilin anlamını, dil bağlamını ve dilin çeşitli biçimlerini içeren bir bilim dalıdır. Bu nedenle, NLP'nin çeşitli önemli terimlerini ve NLP'nin farklı seviyelerini anlamak önem arz etmektedir (Khurana vd., 2023).

- **Fonoloji**: Khurna vd.'ne (2023) göre bu alan, bir dildeki seslerin incelenmesi ile ilgilidir. Dilbilimin seslerin sistemli düzenlemesine atıfta bulunan bölümüdür. "Phonology" terimi, ses veya ses anlamına gelen Yunanca "phono" terimi ile kelime veya

konuşma anlamına gelen "-logy" eki birleşiminden gelir. Fonoloji, herhangi bir insan dilinin anlamını kodlamak için sesin semantik kullanımını içermektedir.

- **Morfoloji**: Kelimelerin oluşumunu inceleyen bir alandır. Kelimenin farklı parçaları, morfem adı verilen anlamın en küçük birimlerini temsil eder. Morfoloji, kelimenin doğası olarak bilinen morfemler tarafından başlatılan bir konuyu içerir. Bir morfem örneği, "precancellation" kelimesinin üç ayrı morfeme ayrılabilir olmasıdır: önek "pre", kök "cancella" ve sonek "-tion". Morfemlerin yorumlanması, tüm kelimelerde aynı kalır, sadece anlamı anlamak için insanlar herhangi bir bilinmeyen kelimeyi morfemlere bölebilirler. Örneğin, bir fiile -ed soneki eklemek, fiilin eyleminin geçmişte gerçekleştiğini ifade eder. Kendi başlarına bölünemeyen ve anlam taşıyan kelimelere "Leksikal morfem" denmektedir (örneğin: masa, sandalye). Leksikal morfemle birleştirilen kelimeler (örneğin: -ed, -ing, -est, -ly, -ful), "Gramatikal morfem" olarak bilinir (örneğin: çalıştı, danışmanlık, en küçük, Muhtemel, Kullanma). Bir araya gelen gramatikal morfemlere "bağlı morfemler" denir (örneğin: -ed, -ing). Bağlı morfemler, eğilimsel morfemler ve türeme morfemler olarak ayrılabilir. Bir kelimeye eğilimsel morfemler eklemek, zaman, cinsiyet, kişi, mod, belirlilik ve canlılık gibi farklı dilbilgisi kategorilerini değiştirir. Örneğin, -ed eğilimsel morfemi eklemek, kök park'ı parked yapar. Türeme morfemleri, bir kelimeyle birleştirildiğinde kelimenin anlamını değiştirir. Örneğin, "normalize" kelimesinde, kök normal ile birleştirildiğinde, bound morfem -ize kelimeyi bir sıfattan (normal) bir fiile (normalize) değiştirir (Khurana vd., 2023).

- **Sözdizimi**: Sözdizimi, cümlelerin düzeni ve yapısıyla ilgilenmektedir. Kelime düzeyinde yapılan etiketleme işleminden sonra, kelimeler gruplara ayrılmakta ve bu gruplar cümleleri oluşturmak için bir araya getirilmektedir, ardından cümleler sentaktik düzeyde birleştirilmektedir. Bu düzey, bir cümlenin dil bilgisel yapısını analiz ederek doğru bir cümle oluşturulmasını sağlamaktadır. Bu düzeyin çıkışı, kelimeler arasındaki yapısal bağımlılıkları ortaya koyan bir cümle olarak meydana gelmektedir. Ayrıca, "parsing" olarak da bilinir ve bireysel kelimelerin anlamından daha fazla anlam taşıyan ifadeleri ortaya çıkarmaktadır. Sentaktik düzey, kelime sırası, durak kelimeler, morfoloji

ve kelimenin PoS (Sözcük Türü) gibi faktörleri inceleyerek cümlenin dilbilim yapısını analiz etmektedir. Bu, kelime düzeyinin dikkate almadığı bir analizdir. Kelime sırasını değiştirmek, kelimeler arasındaki bağımlılığı değiştirecek ve cümlelerin anlaşılmasını etkileyebilecektir. Örneğin, "Köpeğim bir kaza sonucu kaçtı." ve "Bir kaza sonucu köpeğim kaçtı." cümlelerinde sadece söz dizimi farklıdır, ancak farklı anlamlar taşımaktadır. Bu düzey, cümlelerin anlamını değiştireceği için durak kelimeleri korunmaktadır. Kelimeyi temel haline getirmeyi ve kök çıkarmayı desteklemez, çünkü kelimeleri temel formuna çevirmek cümlenin dilbilgisini değiştirebilmektedir (Khurana vd., 2023).

- **Anlambilim**: Bu alan, dil yapılarının, ifadelerin ve metinlerin anlamı ile ilgilenir. Semantik düzeyde, en önemli görev bir cümlenin doğru anlamını belirlemektir. Bir cümlenin anlamını anlamak için insanlar, dil hakkındaki bilgi ve o cümlede bulunan kavramlara güvenirler, ancak makineler bu tekniklere güvenemezler. Semantik işleme, bir cümlenin mantıksal yapısını işleyerek, cümlenin içindeki kelimeler arasındaki etkileşimleri anlamak için en uygun kelimeleri tanımlayarak bir cümlenin olası anlamlarını belirler. Örneğin, bir cümlenin gerçek kelimeleri içermese bile "filmler" hakkında olduğunu anlar, ancak gerçek kelimeleri içermez, ancak "aktör", "aktris", "diyalog" veya "senaryo" gibi ilgili kavramları içerir (Khurana vd., 2023).

- **Pragmatik**: Pragmatik, dilin bağlam içinde anlaşılmasını içerir; konuşmacı niyetleri, sosyal bağlam ve anlayış gibi faktörleri dikkate almaktadır. Belge içeriğinin dışından gelen bilgi veya içeriğe odaklanmaktadır. Konuşmacının ne ima ettiği ve dinleyicinin ne çıkardığıyla ilgilenir, genellikle açıkça belirtilmeyen cümlelerin incelenmesini içerir. Gerçek dünya bilgisi, metinde ne hakkında konuşulduğunu anlamak için kullanmaktadır. Bağlamı analiz ederek, metnin anlamlı bir temsili türetilmektedir. Bir cümle belirli değilse ve bağlam o cümle hakkında belirli bilgiler sağlamıyorsa, pragmatik belirsizlik ortaya çıkmaktadır (Walton, 1996). Metin bağlamının bağlı olduğu duruma göre olarak farklı kişiler metnin farklı yorumlarını yapabilirler, bu da pragmatik belirsizliği ortaya çıkarır. Bir metnin bağlamı, aynı belgenin diğer cümlelerine yapılan

referansları içerebilir, bu da metnin anlaşılmasını ve okuyucunun veya konuşucunun arka plan bilgisini etkilemekte ve metinde ifade edilen kavramlara anlam katmada yardımcı olmaktadır. Semantik analiz genellikle kelimelerin kelime anlamına odaklanırken, pragmatik analiz okuyucuların arka plan bilgilerine dayanarak algıladıkları çıkarılan anlamla ilgilenmektedir. Örneğin, "Saat kaç?" cümlesi semantik analizde "Mevcut saati sormak" olarak yorumlanmaktadır. Ancak pragmatik analizde aynı cümle, "Belirli bir süreyi kaçıran birine karşı duyulan rahatsızlığı ifade etmek" anlamına gelebilmektedir. Temelde, semantik analiz dil bilgisel ifadeler ile anlamları arasındaki ilişkiyi keşfederken, pragmatik analiz dil bilgisel ifadelerin anlayışımızı etkileyen bağlamı incelemeye odaklanmaktadır. Pragmatik analiz, kullanıcıların bağlamsal arka plan bilgisini kullanarak metnin amaçlanan anlamını ortaya çıkarmalarına yardımcı olmaktadır (Khurana vd., 2023).

Metin Akışı (Discourse) Seviyesi

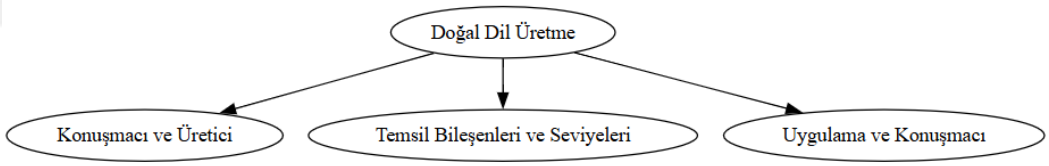
Sözdizimi ve semantik seviye, tek bir cümleyle uğraşırken, NLP'nin metin akışı seviyesi birden fazla cümleyle ilgilenmektedir. Bu seviye, kelimeler ve cümleler arasında bağlantılar kurarak mantıksal bir yapı analizi yapar ve metnin tutarlılığını sağlamaktadır. “Anaphora Çözümleme” ve “Coreference Çözümleme” gibi iki yaygın alt seviye bulunmaktadır. Anaphora çözümlemesi, bir anaforanın atıfta bulunduğu varlığı tanımlayarak metin içinde aynı anlamı taşıyan referansları çözmeyi içermektedir. Örneğin, (i) Ram sınıfta birinci oldu. (ii) O zekiydi. Burada (i) ve (ii) bir araya gelerek bir metin akışı oluşturur. İnsanlar, (ii) içindeki "O" zamirinin "Ram"e atıfta bulunduğunu hızlıca anlayabilmektedirler. Coreference çözümlemesi, bir metinde aynı varlığa atıfta bulunan tüm ifadeleri bulmayı içermektedir. Bu, belge özetleme, bilgi çıkarma gibi yüksek düzeyli NLP görevleri için önem arz etmektedir. Aslında, anafora, co-reference adı verilen bir süreçle kodlanmaktadır (Khurana vd., 2023).

Özetle, Doğal Dil İşleme (NLP), dilin farklı seviyelerinde analiz yaparak makinelerin dil anlamasını sağlar. Fonoloji, bir dildeki seslerin düzenlemesini inceler; Morfoloji, kelimelerin oluşumunu ve anlam birimlerini ele alır. Sözdizimi, cümlelerin

yapısal düzenini analiz eder. Semantik, cümlelerin anlamını belirler. Pragmatik, dilin bağlam içinde anlaşılmasını sağlar. Metin Akışı seviyesi, birkaç cümleyi içeren metinlerin tutarlılığını sağlar ve Anaphora-coreference çözümleme gibi alt seviyeleri içerir. Bu seviyeler, dilin farklı yönlerini kapsayarak dil anlama ve işleme süreçlerini yönetir.

1.3.2 Doğal Dil Üretme

Doğal Dil Üretimi (NLG), anlamlı ifadeler, cümleler ve paragraflar oluşturmayı içeren Doğal Dil İşleme'nin bir parçasıdır. NLG süreci, Şekil 1.3 de görüldüğü gibi hedefleri belirleme, bu hedeflere nasıl ulaşılacağını değerlendirme ve iletişim kaynaklarına dayalı olarak planlama, ardından bu planları metin olarak gerçekleştirme olmak üzere dört aşamada gerçekleşmektedir (Khurana vd., 2023).



Şekil 1.3: Doğal Dil Üretme

Kaynak: Khurana vd., 2023

a) Konuşmacı ve Üretici

Metin oluşturmak için, bir konuşmacı veya bir uygulama ile bir üreteç veya bir programın olması gerekmektedir. Bu, uygulamanın niyetlerini akıcı bir ifadeye dönüştüren ve duruma uygun bir şekilde anlamlı bir cümle oluşturan bir programı ifade etmektedir (Khurana vd., 2023).

b) Temsil Bileşenleri ve Seviyeleri

Khurana vd.'ne (2023) göre dil oluşturma süreci, aşağıdaki birbirine geçmiş görevleri içerir;

• İçerik seçimi: Bilgi seçilmeli ve küme içine dâhil edilmelidir. Bu bilgilerin temsil birimlerine nasıl ayrıştırıldığına bağlı olarak, birimlerin bazı kısımları varsayılan olarak eklenirken diğerleri çıkarılmalıdır.

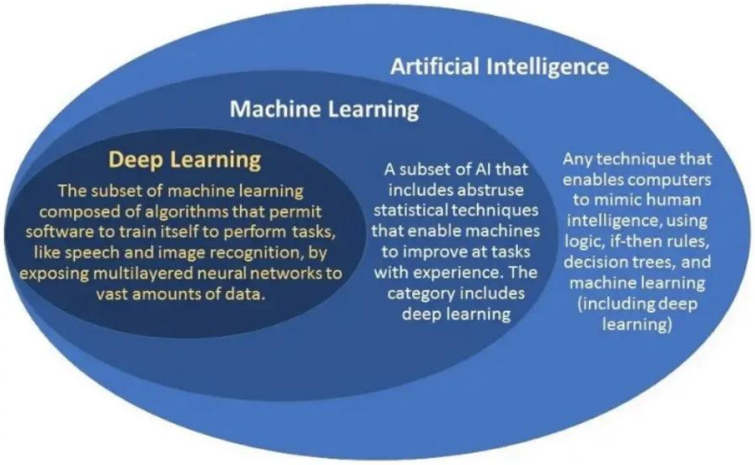
• Metinsel Organizasyon: Bilgi, dilbilgisine göre metinsel olarak düzenlenmelidir; hem sıralı hem de dil bilgisel ilişkiler açısından düzenlenmelidir, örneğin, değişiklikler.

• Dil Kaynakları: Bilginin gerçekleştirilmesini desteklemek için dil kaynakları seçilmelidir. Sonunda, bu kaynaklar belirli kelimelerin, deyimlerin, sözdizimsel yapıların seçimlerine dönüşecektir.

• Gerçekleştirme: Seçilen ve düzenlenen kaynaklar, gerçek bir metin veya ses çıktısı olarak gerçekleştirilmelidir.

c) Uygulama veya Konuşmacı

• Bu, durum modelini sürdürmek için kullanılır. Burada konuşmacı, yalnızca süreci başlatır, dil oluşturmada yer almaz. Konuşmacı, geçmişi depolar, potansiyel olarak ilgili olan içeriği yapılandırır ve bildiklerinin bir temsilini kullanır. Tüm bunlar durumu oluştururken, konuşmacının sahip olduğu önermelerin bir alt kümesini seçer. Tek gereksinim, konuşmacının durumu anlamasıdır (Khurana vd., 2023).



Şekil 1.4: Yapay Zekâ, Makine Öğrenimi, Derin Öğrenme

Kaynak: Unite.AI, 2025

Özetle, yapay zekâ alanındaki önemli alt kollar olan makine öğrenimi ve derin öğrenme arasında güçlü bir ilişki bulunmaktadır. Bu ilişki Şekil 1.4 de gösterilmiştir. Makine öğrenimi, bilgisayar sistemlerinin belirli görevleri yerine getirebilmek için verilerden öğrenmelerini sağlayan bir alan olarak öne çıkar. Derin öğrenme ise bu alanın özel bir alt dalıdır ve genellikle yapay sinir ağları kullanılarak karmaşık yapıdaki verilerin analizi ve öğrenilmesi üzerine odaklanır. Makine öğrenimi, derin öğrenmenin temelini oluşturur. Derin öğrenme algoritmaları, büyük miktarda veri üzerinde karmaşık desenleri belirlemek ve öğrenmek için çok katmanlı yapay sinir ağlarını kullanır. Bu ağlar, insan beyninin çalışma prensiplerine benzer şekilde, katmanlar arasında bilgiyi ileterek ve temsil ederek karmaşık görevleri gerçekleştirir. Örneğin, görüntü tanıma, ses tanıma ve doğal dil işleme gibi uygulamalarda derin öğrenme, yüksek düzeyde başarı elde edebilen karmaşık modeller oluşturabilir. Bu nedenle, makine öğrenimi ve derin öğrenme arasındaki ilişki, veri tabanlı öğrenme alanındaki gelişmelerin itici güçlerinden biridir. Makine öğrenimi, algoritmaların geniş bir veri yelpazesinde öğrenme yeteneği sağlarken, derin öğrenme, bu öğrenme sürecini daha karmaşık ve katmanlı modellerle gerçekleştirme kapasitesini temsil eder. İkisi bir araya geldiğinde, yapay zekâ sistemleri, daha önce zor veya manuel olarak çözülen görevleri daha etkili bir şekilde gerçekleştirebilir ve karmaşık ilişkileri anlamak için daha güçlü araçlara sahip olabilir.

2 DERİN ÖĞRENME ALGORİTMALARI

Son yıllarda derin öğrenme algoritmaları, karmaşık ve büyük ölçekli verilerden öğrenme ve tahmin yapma yetenekleri nedeniyle önemli bir popülerlik kazanmıştır. Bu algoritmalar, desen tanıma ve öngörü modelleme gibi görevlerde üstün performans sergileyerek bilgisayar görüşü, doğal dil işleme, robotik ve otonom sürüş gibi çeşitli alanlarda uygulamalar için uygun hale gelmiştir. Derin öğrenme algoritmalarının bu alanlarda kullanımı, birçok durumda insan performansını aşan dönüşümsel atılımlara yol açmıştır. Derin öğrenmenin geleneksel makine öğrenimi algoritmalarına göre üstünlüğü kanıtlanmıştır. Derin öğrenme algoritmaları, bilgisayar görüşü ve doğal dil işleme gibi

geleneksel yapay zekâ uygulamalarında önemli deneysel başarılar elde etmiş temsil öğreniminin özel durumlarıdır. Bu algoritmalar, otomatik olarak daha soyut özellikleri keşfetmelerine izin veren çok seviyeli temsil öğrenmeyi gerçekleştirme yeteneğine sahiptir. Derin öğrenme algoritmaları, bilgisayar görüşü, robotik, doğal dil işleme ve konuşma işleme dâhil olmak üzere çeşitli alanlarda devrim yapmıştır (Bengio vd., 2021).

2.1 Recurrent Neural Network (RNN)

Son yıllarda, sinir ağları üzerine yapılan araştırmalardaki ilgi önemli ölçüde artmaktadır. Bu çaba, beyin tarzında hesaplama, bağlantılı mimariler, paralel dağıtımlı işleme sistemleri, nöromorfik hesaplama, yapay sinir sistemleri gibi birçok farklı şekilde adlandırılmaktadır. Bu çabaların ortak teması, beynin geleneksel seri bilgisayarlardan oldukça farklı bir hesaplama cihazı modeli olarak incelenmesine olan ilgidir (Rumelhart vd., 1994).

Strateji, beyin benzeri sistemlerin basitleştirilmiş matematiksel modellerini geliştirmek ve ardından bu modelleri inceleyerek bu tür cihazlar aracılığıyla çeşitli hesaplama problemlerinin nasıl çözülebileceğini anlamaktır. Bu çalışma, çeşitli disiplinlerden bilim insanlarını çekmektedir. Sinir devrelerini belirli hayvanların beyinlerinde modellemeye çalışan nörobilimciler, beyin benzeri sistemlerin dinamik davranışlarıyla fizikteki nonlineer dinamik sistemler arasındaki benzerlikleri gören fizikçiler, beyin benzeri bilgisayarlar üretmeye ilgi duyan bilgisayar mühendisleri, biyolojik organizmaların zekâsı üzerine çalışan yapay zekâ (YZ) uzmanları, pratik problemleri çözmeye çalışan mühendisler, insan bilgi işleme mekanizmalarıyla ilgilenen psikologlar, bu tür sinir ağı sistemlerinin matematiksel yönleriyle ilgilenen matematikçiler, bu sistemlerin zihnimizle ve beyinle olan ilişkisini nasıl değiştirdiğine dair düşünceler üreten filozoflar ve diğer birçok bilim insanı bu çalışma alanında faaliyet gösteren bilim insanları olarak yer almaktadır. Bu geniş ilgi çemberi ve yetenek yelpazesi, bu alanı parlak genç öğrenciler için bir cazibe merkezi haline getirmiştir (Rumelhart vd., 1994).

Rumelhart vd.'ne (1994) göre sinir ağları detayları değişse de, en yaygın modeller genellikle nöronu temel işlem birimi olarak almaktadır. Her bir işlem birimi, bir aktivite seviyesi (bir nöronun polarizasyon durumunu temsil eden), bir çıkış değeri (bir nöronun ateşleme hızını temsil eden), bir dizi giriş bağlantısı (hücre üzerindeki sinapsları ve dendritini temsil eden), bir bias değeri (bir nöronun iç dinlenme seviyesini temsil eden) ve bir dizi çıkış bağlantısı (bir nöronun aksonal projeksiyonların temsil eden) ile karakterize edilmektedir. Bu birimin her bir yönü matematiksel olarak gerçek sayılarla temsil edilmektedir. Bu nedenle, her bir bağlantının, birimin aktivasyon seviyesi üzerindeki girişin etkisini belirleyen bir ağırlığı (sinaptik güç) vardır. Bağlantıların etkisi genellikle pozitif (uyarıcı) veya negatif (engelleyici) olabilmektedir. Giriş hatları lineer olarak toplandığı varsayılır ve bu durum, i birimi için t zamanındaki aktivasyon değerini aşağıdaki Şekil 2.1 de yer alan formülle ifade edilmektedir.

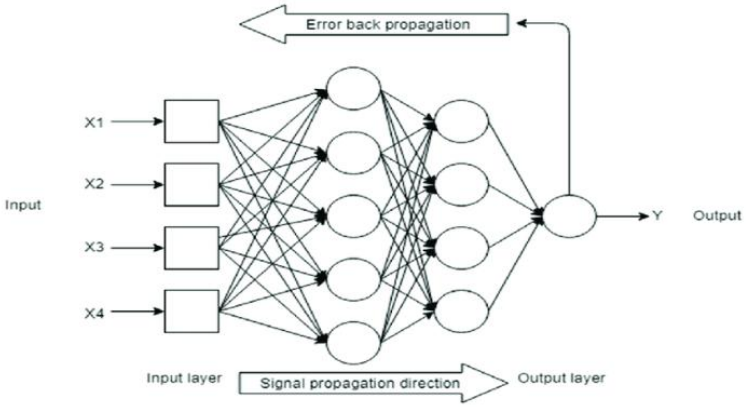
$$a_i(t) = \sum_j w_{ij} x_j + p_i$$

Şekil 2.1: Nöron Aktivasyon Değeri

Kaynak: Rumelhart vd., 1994

Burada w_{ij}, j nöronundan i nöronuna olan bağlantının gücü, p_i nöronun bias değerini, x_j ise j nöronunun çıktı değerini temsil etmektedir. Her nöron t zamanında bir aktivasyon değeri taşımakta, bu değer diğer nöronlara çıktı verip vermeyeceğini belirlemektedir (Rumelhart vd., 1994).

Yapay sinir ağlarında öğrenme probleminin temelinde, ağın istenilen hesaplamayı gerçekleştirebilmesini sağlayacak bir bağlantı gücü kümesini bulma zorluğu yatmaktadır. Bu bağlantı gücü, şu anda en popüler öğrenme sistemlerinden biri olan ve neredeyse tüm uygulamaların temelini oluşturan geri yayılımdır (backpropagation). Tipik ağ yapısı Şekil 2.3 de gösterilmiştir.



Şekil 2.3: Derin Öğrenme Geri Yayılım

Kaynak: Azlah vd., 2019

Giriş birimlerinden bir dizi gizli birim aracılığıyla çıkış birimlerine bağlanan bir yapı mevcuttur. Genel durumda, gizli birim ve birimler arasında bağlantıların sayısı ve yapılandırması herhangi bir sayıda ve yapıda olabilir. Genellikle, gizli birimler bir dizi gizli birim katmanı olarak yapılandırılmakta, çoğu zaman bir tek katman gizli birim bulunsa da, bazı uygulamalarda iki veya daha fazla katman gizli birimin bulunması uygun olabilmektedir. Ağa bir örnek giriş/çıkış çifti kümesi (bir eğitim seti) sağlanır ve ağ, giriş/çıkış çiftlerini çekmiş olduğu fonksiyona ulaşmak için bağlantıları (ağırlıkları) değiştirmektedir. Ardından ağın genelleme yeteneği test edilir. Hata düzeltme öğrenme yönergeleri kavramsal olarak oldukça basittir. Prosedür şu şekildedir: Eğitim sırasında bir giriş ağa verilir ve ağ üzerinde dolaşarak çıkış birimlerinde bir değer kümesi oluşturur. Ardından gerçek çıkış, istenen hedefle karşılaştırılır ve bir eşleşme hesaplanır. Eğer çıkış ve hedef eşleşiyorsa, ağa herhangi bir değişiklik yapılmaz. Ancak, çıkış hedeften farklıysa, bağlantılardan bazılarında bir değişiklik yapılmalıdır. Sorun, hatanın hangi bağlantılardan kaynaklandığını belirlemektir – bu süreç, kredi atama (veya belki daha iyi bir ifadeyle, suç atama) problemi olarak adlandırılmaktadır. Bağlantılardaki hatanın bu problem için çözümü, gizli katmanları olmayan ağlar için bir süredir bilinmekteydi, ancak bu, genel olarak zor bir problemdir ve tatmin edici bir çözümün eksikliği, daha önce sinir

ağı sistemlerine olan ilginin kaybolmasındaki ana etkenlerden biriydi. 1980'ler, bu soruna oldukça basit ancak güçlü bir çözümün geliştirilmesine yol açtı. Temel fikir, sistemin genel performansını ölçmek ve ardından yapay zekânın bu performansı optimize etmek için bir dizi değişiklik yapmasına izin vermektir. Böylece tanım kümesi ve değer kümesi arasındaki fonksiyona ulaşmak mümkün olmaktadır (Rumelhart vd., 1994).

RNN'ler tekrar eden doğası nedeniyle her veri için aynı işlemi gerçekleştiren Rekürren Yapay Sinir Ağları olarak bilinen bu sinir ağları, metin, zaman serileri, finansal veriler, konuşma, ses, video gibi sıralı veriler için ideal bulunmuş ve doğal dil işleme (NLP) alanında da kullanılmıştır. Bir RNN aslında tamamen bağlı bir sinir ağıdır ve bazı katmanlarının bir döngü haline getirilmesini içermektedir. Bu döngü genellikle iki girişin toplamı veya birleşimi üzerinde bir iterasyon, bir matris çarpımı ve bir doğrusal olmayan fonksiyon içermektedir (Thomas, 2019).

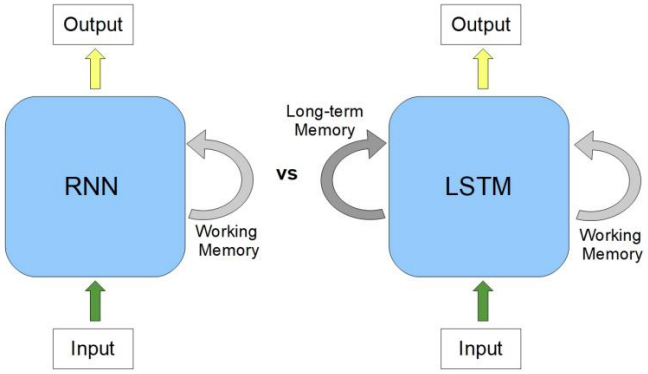
Thomas (2019)'a göre RNN'lerin etkili bir iç hafızası vardır ve önceki girdilerin sonraki tahminleri etkilemesine izin verir. Eğer önceki kelimelerin ne olduğunu bilinirse, bir cümledeki bir sonraki kelimeyi daha doğru bir şekilde tahmin etmek çok daha kolay olacaktır. Genellikle RNN'ler için uygun olan görevlerde, öğelerin dizisi, önceki öğeden daha önemli veya en az onun kadar önemli olmaktadır.

Özetle Rekürren Yapay Sinir Ağları (RNN'ler), sinir ağlarının bir alt türüdür ve özellikle sıralı verilerin işlenmesi için tasarlanmıştır. RNN'ler, bir döngü içinde yapılandırılmış ve her adımda önceki adımların bilgilerini içeren bir yapıya sahiptir. Bu özellik, RNN'leri zaman serileri, metin ve diğer sıralı veri türlerini işleme konusunda etkili kılmaktadır. RNN'lerin temel avantajlarından biri, önceki girdilerin sonraki tahminlere etki edebilmesidir. Bu özellik, dil modellemesi, metin üretimi, zaman serisi analizi gibi birçok uygulama için ideal bir çözüm olarak görülmektedir.

2.2 Long Short-Term Memory (LSTM)

Derin öğrenme modelleri, son zamanlarda yapay zekâ ve makine öğrenme alanlarını, makinelerin büyük miktarda veriden öğrenmesine ve kesin tahminlerde bulunmasına olanak tanıyarak dönüştürmüştür. Long Short-Term Memory (LSTM) ağı, sıralı girişleri işlemek üzere tasarlanmış bir Rekurrent Sinir Ağı (RNN) türü olarak, en etkili derin öğrenme modellerinden biri olarak öne çıkmaktadır. LSTM ağları, bankacılık, konuşma tanıma, görüntü ve video analizi ile doğal dil işleme de dâhil olmak üzere bir dizi endüstride yaygın olarak kullanılmaktadır. Bunun nedeni, düzenli RNN'leri etkileyen kaybolan gradyan problemine karşı dirençli olmaları ve dil bağlamı gibi sıralı veride uzun vadeli ilişkileri yakalama yetenekleridir (Kanrar vd., 2023).

Geriye Yayılım tekniği, kaybolan gradyan problemi ile karşılaşmaktadır, bu da gradyanların çok küçük olması nedeniyle öğrenmenin yavaş veya hiç gerçekleşmediği bir soruna yol açmaktadır. Bu durum, ağırlıkları güncellemek için gradyanların çok küçük olması sebebiyle ortaya çıkmaktadır. Bu da ağın uzun vadeli bağımlılıkları öğrenmekte zorluk yaşamasına neden olmaktadır. LSTM'ler, bu sorunu aşmak için özel olarak tasarlanmıştır; uzun süre bilgi saklayabilen bir bellek hücresi içermeleri, özellikle doğal dil işleme, konuşma tanıma ve zaman serisi analizi gibi sıralı verilerle ilgili görevler için son derece etkili olmalarını sağlamaktadır. LSTM modelinde, bellek hücresi, her zaman adımında kullanılan bir dizi kapı mekanizması aracılığıyla güncellenmektedir. İki ana kapı olan giriş kapısı ve çıkış kapısı, bilgi akışını düzenleme, hücreden bilgi çıkarma ve eski bilgileri unutma görevlerinden sorumlu olmaktadır. Bu kapılar, hücredeki bilgi akışını kontrol ederek, modelin uzun vadeli bağımlılıkları daha etkili bir şekilde öğrenmesini sağlamaktadır (Kanrar vd., 2023).



Şekil 2.4: RNN ve LSTM Mimarileri

Kaynak: Yasrab vd., 2020

Şekil 2.4'den de anlaşılacağı gibi RNN'ler giriş dizililerini işleme koymak için belleklerini kısa süreli kullanırlar ancak LSTM'ler RNN'lere ek olarak uzun vadede kullanabilecekleri belleğe sahiptirler ve böylece geçmiş veriyi hatırlayabilirler.

LSTM modelleri, yaygın uygulamalarına rağmen, geniş bir değişiklik ve yapılandırma yelpazesi ile karmaşık ve yüksek özelleşmiş bir araştırma alanı olmayı sürdürmektedir. Ayrıca, her görev için tek bir yaklaşım bulunmamaktadır, bu nedenle uygun mimariyi seçmek, görevin ihtiyaçlarının, kullanılan verinin ve mevcut hesaplama kaynaklarının dikkatli bir analizini gerektirir. Ayrıca, her iş için tek bir yaklaşım bulunmamakta ve uygun mimariyi seçmek, görevin gereksinimlerinin, kullanılan verinin ve mevcut hesaplama kaynaklarının dikkatli bir analizini gerektirmektedir (Kanrar vd., 2023).

Ancak, LSTM modellerinin bazı sınırlamaları da vardır. Bunlardan biri, hesaplama karmaşıklıklarıdır, bu da onları bazı diğer makine öğrenimi modellerine kıyasla daha yavaş eğitilebilir hale getirmektedir. Ayrıca, LSTM modelleri sınırlı eğitim verisiyle karşılaştığında aşırı öğrenme sorunu yaşayabilmekte ve karmaşık içyapısı nedeniyle yorumlanmaları zor olabilmektedir (Kanrar vd., 2023).

Özetle diğer makine öğrenimi modellerine kıyasla, LSTM modellerinin birkaç avantajı bulunmaktadır. İlk olarak, dizi içeren verilerde uzun vadeli bağımlılıkları yakalayabilirler ki bu, birçok makine öğrenimi görevi için kritiktir. İkinci olarak, değişken uzunluktaki veri dizilerini işleyebilirler, bu da onları konuşma tanıma ve makine çevirisi gibi görevler için uygun hale getirmektedir. Son olarak, LSTM modelleri, geri yayılım tekniğini kullanılarak uçtan uca eğitilebilirler, bu da onları daha geniş makine öğrenimi süreçlerine bütünleşmiş yapılanmayı kolaylaştırmaktadır.

2.3 Transformers

RNN'ler (LSTM, GRU), rekürrent bir alt yapıya sahiptir ve tanım gereği rekürrendirler. Bu temel özelliğin varlığı nedeniyle öğrenme sürecini paralelleştirmek zor olmaktadır. Rekürrent modeller, genellikle giriş ve çıkış dizilerinin sembol pozisyonları boyunca hesaplamayı bölmektedir. Hesaplama zamanındaki adımlara pozisyonları hizalayarak, önceki gizli durum h_{t-1} ve t pozisyonu için girişin bir fonksiyonu olarak bir dizi gizli durum h_t üretirler. Bu doğal olarak sıralı doğa, eğitim örnekleri içinde paralelleştirmeyi engeller; bu, bellek kısıtlamalarının örnekler arasında toplu işlemleri sınırladığı için uzun dizin uzunluklarında kritik hale gelmektedir. Bir başka deyişle transformerlar cümlenin uzun menzilli bağımlılıklarını yakalamak ve paralelleştirilebilir olmak avantajına sahip olmaktadırlar (Gillioz vd., 2020).

Gillioz vd.'ne (2020) göre transformerlar, bir kodlayıcı-çözücü yapısına dayanmaktadırlar. Transformer ağ modeli burada bir dizi $X = (x_1,...,x_N)$ alır ve bir gizli temsil $Z = (Z_1,..., Z_N)$ dizisi üretir. Bu üretilen dizi giriş dizisinin içsel veya gizli temsilini ifade etmektedir. Bu modelin otomatik regresif özelliği nedeniyle her bir $Y_M=(y_1,\ldots,y_M)$ bir elemanı sırasıyla üretir. Y_M'yi tahmin etmek için $Y_{M-1}=(y_1,\ldots,y_{M-1})$ ve girdinin gizli temsil olan Z yi kullanır. Kodlayıcı ve çözücü aynı çok-başlı dikkat mekanizmasını kullanmaktadır.

Tay vd.'ne (2021) göre Transformer modellerinin başarısının temelinde, sorgu-anahtar-değer nokta çarpımı dikkat mekanizması yatmaktadır. Bu tam bağlantılı (fully

connected) token grafikleri, uzun menzilli bağımlılıkları modelleyebilen ve güçlü bir tümevarımsal önyargı sağlayabilen bu öz-dikkat mekanizmasına geniş bir şekilde atfedilmektedir.

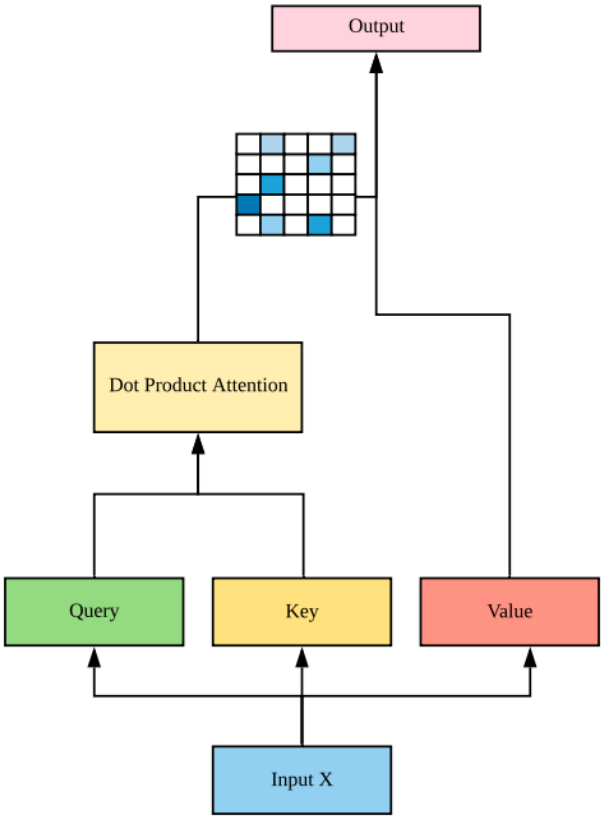
Kodlayıcı (encoder), bir giriş dizisini (örneğin, bir dildeki bir cümleyi) adım adım işlemektedir. Giriş dizisindeki bilgileri bir temsile dönüştürür. Ardından bu temsili, cümlenin anlamını yakalamak için önemli olan anahtar kelimelerle zenginleştirmektedir. Bu önemli bilgileri içeren temsil, ardından çözücüye (decoder) iletilmekte ve çözücü, bu temsil ve çeviri göreviyle birlikte alınan anahtar kelimeleri kullanarak çıkış dizisini oluşturmaktadır. Bu sayede çözücü, çeviriyi daha iyi anlayarak ve doğru bir şekilde üretebilmektedir. Transformer modelinin iki tür dikkati vardır: self-attention (cümle içindeki kelimeler arasındaki bağlantı) ve Encoder-Decoder attention (kaynak cümledeki kelimeler ile hedef cümledeki kelimeler arasındaki bağlantı). (Kublik ve Saboo, 2022).

Dikkat mekanizması bir giriş dizisindeki her bir elemanın çıkışa katkısının belirleme sürecidir. Softmax aktivasyon fonksiyonu kullanılarak hesaplanır. Fonksiyon aşağıdaki gibidir.

$$\text{Attention}(Q,K,V)=\text{Softmax}(Q\,K^{T}/\sqrt{d^{k}})V$$

K anahtar girişi temsil eden vektördür. Her giriş elamanı için bir anahtar vektör oluşturulur. Anahtarlar, modelin öğrenme sürecinde verinin belirli özelliklerini temsil etmek için kullanılır ve skor hesaplamasında ve dikkat ağırlıklarının belirlenmesinde önemli bir rol oynarlar. Sorgu (Q), dikkat mekanizmasının odaklanmak istediği konuyu temsil eden bir vektördür. Başka bir ifadeyle metnin anlamını temsil eden vektördür. Her giriş elemanı için bir sorgu vektörü oluşturulur. Sorgular, modelin belirli bir giriş elemanına ne kadar dikkat etmesi gerektiğini belirlemek için kullanılır. Skor hesaplamasında ve dikkat ağırlıklarının belirlenmesinde kullanılır. Değer (V), giriş elemanının temsil edildiği bir vektördür. Her giriş elemanı için bir değer vektörü oluşturulur. Değerler, sorgu ve anahtardan elde edilen dikkat ağırlıkları ile ağırlıklı bir şekilde toplanarak çıkışı oluşturmak için kullanılır (Tay vd., 2021).

Dikkat mekanizmasının mimarisi aşağıdaki Şekil 2.5 de verilmiştir.



Şekil 2.5: Transformer Dikkat Mekanizması Mimarisi

Kaynak: Tay vd.,2021

Örneğin, "The cat sat on the mat once it ate the mouse." cümlesine bakılırsa, cümledeki "it" ifadesi "the cat" mi yoksa "the mat" ilişikli olduğu açık değildir. Transformer modeli, "it" ifadesini "the cat" ile güçlü bir şekilde ilişkilendirebilir. İşte bu dikkat (attention) mekanizmasıdır (Kublik ve Saboo, 2022).

Özetle transformerlar geniş ve çeşitli dil verileri üzerinde eğitilerek oluşturulmuş modellerdir. Bu modellere “pre-trained model” de denir. Bu modeller oluşturulduktan sonra belirli görevler için uyarlanırlar. Dikkat mekanizması, sorgu ve anahtar arasındaki

benzerliklere dayanarak, giriş elemanlarının belirli özelliklerine odaklanabilir. Ağırlıklı değerlerin toplanmasıyla çıkış elde edilir ve bu mekanizma, özellikle uzun ve bağlamsal zengin verilerin işlenmesinde, modelin belirli bölgelere odaklanarak daha etkili öğrenme yapmasına olanak tanımaktadır.

2.4 BERT (Bidirectional Encoder Representation from Transformers)

Doğal metin işleme görevleri iki kategoride incelenir: (i) bütünsel ve (ii) belirtilmiş görevler. Bütünsel görevler metni cümle düzeyinde işler. Açıkça ifade edilirse, bir metni bir bütün olarak ele alır ve genel anlamını çıkarmaya çalışır. Metin özetleme, metin sınıflandırma, duygu analizi gibi işlemler bütünsel görevlere verilebilecek örneklerdir. Belirtilmiş görevler (Tokenized görevler) metni daha küçük öğelere (belirteç ve kelimelere) ayırır ve her bir parça üzerinde özel analizler yapmaktadır. Her iki kategori de son zamanlarda önceden eğitilmiş modelleri kullanmaktadır ve bu da özel modellerin tasarlanması ve eğitilmesi için geçen süreyi önemli ölçüde azaltabilirken yüksek bir etkinlik seviyesini sürdürebilmektedir (Dai ve LE, 2015).

Derin metin temsil modellerini önceden eğitme yaklaşımları konusunda iki ana kategori vardır: (i) özellik çıkarma ve (ii) model ince ayarı. Her iki yaklaşım da aynı ön öğrenme amaç fonksiyonlarını ve tek yönlü metin analizini kullanır. İlk yöntem göreve özel mimariye sahip modelleri kullanarak belirli bir temsili eğitmektir. Bu temsil, ardından uygulamalı modellerde ek özellik olarak kullanılır. Bu modele örnek olarak ELMo gösterilebilir. İkinci yaklaşım, belirli bir göreve özgü parametreleri kullanmayan modellerin oluşturulmasını içerir, ancak modelin tüm iç parametrelerini ayarlayarak modelin tekrar eğitilmesi gerekmektedir. Bu yaklaşım, örneğin OpenAI GPT modelinde görülmektedir (Radford, 2018, akt. Dai ve Le, 2015).

BERT modelinin mimarları, mevcut yaklaşımların önemli bir sınırlamasının odak noktası olduğunu ve bu durumun mümkün olan sinir ağı mimarilerinin seçimini daralttığını belirtiyor. Örneğin, GPT modeli, bir kelimenin temsilinin sadece önceki, ancak sonraki belirteçleri dikkate almayan soldan sağa metin taramalarını kullanır. Bu,

bütünsel metin analizi için en uygun olmayabilir, ancak belirtilmiş analiz için bu yaklaşım, kelimenin anlamının sol ve sağ bağlamına bağlı olabileceğinden modelin anlamsal gücünü önemli ölçüde azaltmaktadır (Dai ve LE, 2015). BERT, bu sınırlamayı aşmak için "maskeli dil modelleri" adı verilen öğrenme yöntemini kullanarak yapmaktadır. Başka bir ifadeyle, belirli bir temsilin öğrenilmesinin hedef fonksiyonu, bir metinde rastgele seçilmiş ve maskelenmiş bir kelimeyi tahmin etme görevini ifade ederek ve yalnızca çevresel bağlamı dikkate almaktadır. Bu şekilde, derin çift yönlü bir transformer eğitilmektedir (Dai ve LE, 2015).

BERT modelinin eğitim süreci, etiketlenmemiş veriler üzerinde ön eğitim ve belirli bir uygulama problemi için etiketli veriler üzerinde ek eğitimi içerir. Göreve bağlı olarak, tekrarlama süreci ve kullanılan mimariler farklılık gösterebilir, ancak hepsi aynı model ve aynı parametre setine dayanır (Dai ve LE, 2015).

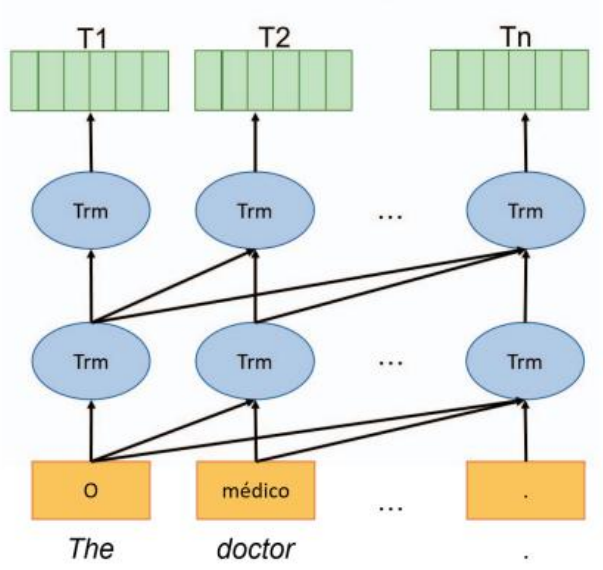
GPT (Generative Pre-trained Transformer) ve GPT-2 aslında eğitim sırasında soldan sağa bağlam kullanan kendi çıktısını ardışık olarak üreten (otoregresif) dil modelleridir. Diğer taraftan, BERT çift yönlü olarak tasarlanmıştır. GPT modelleri bir dizideki önceki kelimeleri kullanarak bir sonraki kelimeyi tahmin ederken, BERT hem soldan hem de sağdan bağlamı çift yönlü olarak dikkate alarak eksik kelimeleri tahmin etmeyi öğrenir. BERT'in bu çift yönlü eğitimi, bir cümlenin tam bağlamını anlamanın kritik olduğu görevler için potansiyel olarak daha etkili olmasını sağlamaktadır. BERT, bir cümlenin sol ve sağ bağlamını birleştirerek çift yönlü bir temsil sağlayabilir ve akıl yürütme görevleri için daha iyi bir bağlam çıkarıcısına olanak tanımaktadır. GPT ve GPT-2 gibi, BERT'in de dil temsilini öğrendiği bir gözetimsiz ön eğitim aşaması içermektedir. Bununla birlikte, doğal çift yönlü mimarisi nedeniyle standart Dil Modeli hedefini kullanarak eğitilemez. Aslında, BERT'in çift yönlülüğü, her kelimenin kendisini görmesine izin verir ve bu nedenle kolay şekilde bir sonraki belirteci tahmin edebilir. Ön eğitim aşaması bittiğinde, modelin alt görevlere ayarlanması gerekmektedir. BERT'in Transformer mimarisi sayesinde, alt görevler işlemler doğrudan yapılabilir çünkü aynı

yapı hem ön eğitim hem de ince ayarlama için kullanılmaktadır. Yalnızca alt görevin gereksinimlerine uyacak şekilde son katmanı değiştirmek yeterli olmaktadır (Gillioz vd., 2020).

2.5 GPT (Generic Pre-trained Transformer)

GPT metin üreten bir model olarak geliştirilmiştir. Üretken (generatif) modelleme, istatistiksel modellemenin bir dalı olarak bilinmektedir. Bu, matematiksel olarak dünyayı yaklaşık bir şekilde temsil etmenin bir yöntemi olarak ifade edilmektedir. Fiziksel ve dijital dünyalarda kolayca erişilebilen inanılmaz miktarda bilgi ile çevrilmekteyiz. Zor olan, bu veri hazinesini analiz edip anlayabilen zeki modeller ve algoritmalar geliştirebilmektir. Üretken modeller, bu hedefe ulaşmak için en umut verici yaklaşımlardan biridir (Kublik ve Saboo, 2022).

Bir modeli eğitmek için, modelin belirli bir görevi gerçekleştirmeyi öğrenmesine yardımcı olan örneklerin bir koleksiyonu olan bir veri seti hazırlanmalı ve ön işleme yapılması gerekmektedir. Genellikle, bir veri seti belirli bir alan içinde büyük miktarda veri içermektedir. Bir modelin bir arabayı ne olduğunu öğrenmesi için milyonlarca araba resimlerinin oluşturduğu veri kümesi buna örnek verilebilir. Veri setleri aynı zamanda cümlelerin veya ses örneklerinin şeklini alabilir. Modeli birçok örnek ile tanıttıktan sonra, benzer veri üretebilmesi için eğitilmesi gerekmektedir (Kublik ve Saboo, 2022).



Şekil 2.6: OpenAI GPT Pre-Training Mimarisi

Kaynak: Schneider vd., 2021

Şekil 2.6 'dan da anlaşılacağı gibi bu mimari yaklaşım, büyük bir külliyat kullanarak transformer çözücüleri soldan sağa doğru eğitilmesi gerekmektedir.

Özetle, GPT (Generative Pre-trained Transformer), derin öğrenme alanında bir algoritmadır. GPT, özellikle doğal dil işleme görevlerinde başarıyla kullanılan bir dil modelidir. Transformer mimarisini kullanır ve önceden eğitilmiş bir modeldir, yani büyük bir dil veri kümesi üzerinde öğrenilmiş ve daha sonra çeşitli doğal dil işleme görevlerine uyarlanabilmektedir. GPT 'in ana özelliği, dil anlama ve üretme yeteneklerini içeren büyük dil modellerini başarıyla oluşturabilmesidir. Bu modele, belirli bir görev için özel olarak eğitilmeden önce geniş bir dil anlama yeteneği kazandırılmaktadır. GPT serisi, OpenAI tarafından geliştirilmiştir ve özellikle GPT-3, şu ana kadar geliştirilen en büyük ve güçlü dil modellerinden biridir.

2.6 ERNIE (Enhenced Representation through Knowledge Integration)

ERNIE, BERT'in maskeleme stratejisinin ilham aldığı, bilgi maskeleri stratejileri ile geliştirilmiş bir dil temsil modeli olarak bilinir. Bu strateji, varlık düzeyinde maskeleme ve ifade düzeyinde maskeleme içermektedir. Varlık düzeyi stratejisi, genellikle birden fazla kelime içeren varlıkları maskeleyerek çalışır. İfade düzeyi stratejisi ise bir kavram birimi olarak bir araya gelen birkaç kelimenin tamamını maskeleyerek çalışır. Deneysel sonuçlar, ERNIE'nin diğer temel yöntemlere göre daha iyi performans sergilediğini ve doğal dil işleme görevlerinde yeni state-of-the-art sonuçlar elde ettiğini göstermektedir. Bu görevler arasında doğal dil çıkarımı, semantik benzerlik, adlandırılmış varlık tanıma, duygu analizi ve soru cevaplama bulunmaktadır (Sun vd., 2019).

Özetle:

- **Derin Öğrenme Algoritmaları**: Büyük veriyi işleme kapasiteleri ve insan performansını aşan başarılarıyla NLP, bilgisayar görüşü ve robotik gibi alanlarda çığır açmaktadır.

- **RNN**: Sıralı veri (metin, ses, video) işlemek için uygundur; önceki girişler, sonrakileri etkileyerek bağlamın korunmasını sağlamaktadır.

- **LSTM**: RNN'nin geliştirilmiş bir türü olup, uzun vadeli bağımlılıkları öğrenmede başarılıdır. Bellek hücreleri ve kapılar sayesinde dil bağlamında derin ilişkileri öğrenmektedir.

- **Transformers**: Paralel işleme yeteneğiyle dilin uzun bağımlılıklarını öğrenebilmektedir. Sorgu-anahtar-değer yapısıyla dikkat mekanizması kullanarak, eğitim süresinde avantaj sağlamaktadır.

- **BERT**: İleri-geri yönlü dikkat kullanarak bağlamın her iki tarafından bilgi alarak, NLP görevlerinde üstün performans sağlamaktadır.

3. PARAMETRELER

3.1 Tokenization

Tokenizasyon, metni kelimeler gibi daha küçük birimlere (tokenlara) ayırma sürecidir. Bu işlem, Doğal Dil İşleme (NLP) uygulamaları için kritik bir ön işleme adımıdır. Duygu analizi, soru yanıtlama, makine çevirisi ve bilgi edinme gibi birçok alanda önemli bir rol oynamaktadır. Modern NLP modelleri metni alt kelime birimlerine tokenize etmektedir. Bu alt kelime birimleri, kelimeler ile karakterler arasında bir orta nokta oluşturur ve morfemler gibi dilbilimsel anlamı korurken, nispeten küçük bir kelime dağarcığı ile bile kelime dışı durumları azaltmaya yardımcı olmaktadır (Song vd., 2020).

Girdi metni ön işleme alındıktan sonra, dil işleme sistemlerinin metin olarak değerlendireceği karakter dizisinden oluşan bir dizi elde edilir. Bu dil işleme sürecinin bir aşamasında, metnin unsurları belirli bir sözdizimsel sınıfa ait olarak kabul edilir. Örneğin, "köpek" dizesi tekil bir isim olarak sınıflandırılır. Sınıfların dizeye atanabilmesi için, orijinal metnin (uzun bir dizi olarak düşünülebilir) sınıf üyeleri olarak tanınacak birimlere bölünmesi gerekmektedir. Tokenizasyonun geleneksel bir rolü, bu birimlerin tanınmasını sağlamaktır. Tokenizasyonun diğer geleneksel rolü ise cümle sınırlarını tanımaktır; çünkü çoğu dil analiz aracı cümleyi işleme birimi olarak kabul etmektedir. Cümleler noktalama işaretleri ile bitmektedir. Ünlem işareti ve soru işareti, bu tür noktalama işaretlerinin neredeyse her zaman net örnekleridir. Ancak, nokta son derece belirsiz bir noktalama işaretidir. Bir noktanın tam duraklama mı, bir kısaltmanın parçası mı yoksa her ikisi mi olduğunu belirlemek kolay değildir. Brown külliyatında 48,885 cümle bulunmakta ve bunlardan 3,490'ı (her 14 cümleden biri) en az bir tane terminal olmayan nokta içermektedir. Kelime ve cümle sınırlarını ayırmak, belirsiz noktalama işaretlerinin kullanımını çözmeyi gerektirir. Tokenizasyonun ikinci rolü, bu durumda ele alınması gereken ilk roldür. Yapısal olarak tanınabilir bazı tokenler, sayılar, alfanümerik referanslar, tarihler, kısaltmalar ve noktalama işaretleri gibi belirsiz noktalama işaretleri içermektedir. Bu sınıflardan bazıları, tokenlerin yapısını öngören düzenli ifade dilbilgileri

aracılığıyla tanınabilir. Bu birimler tanındıktan sonra, ayırıcıların yalnızca belirsiz olmayan kullanımları kalır ve bu nedenle kelimeleri ve cümleleri kesin bir şekilde ayırmak için kullanılabilir (Grefenstette ve Tapanainen, 1994).

Tokenizasyon sürecinde dikkat edilmesi gereken husus, bir token ile bir dizi sınıf arasında birebir bir ilişki olup olmadığı veya bir token'ın bir dizi sınıfa karşılık gelip gelemeyeceğidir. Örneğin, Brown külliyatında "governor's" kelimesi tek bir token olarak kabul edilir ve sahiplik durumu olarak etiketlenir. Ancak Susanne külliyatında aynı dize, "governor" ve "'s" olarak iki token'a ayrılır ve her biri kendi etiketine sahiptir. Bu durumda, bir veya iki token seçimi çok önemli görünmese de, sonraki dil işleme aşamalarının yine de bir token ile üretilen sahiplik yapısını yeniden inşa edeceği varsayılabilir. Ham metni dilsel bir işleme hazırlama süreci birçok sorunu beraberinde getirir. Mümkün olduğunca esnekliği sağlamak için tokenizasyon süreci, metnin seçici bir şekilde geçirilebileceği modüler filtreler dizisi olarak ele alınmalıdır. Tokenizasyonun ana amaçlarından biri, cümle ve kelime sınırlarını tanımak, böylece sözlük arama işlemlerinin devam edebilmesini sağlamaktır. Bazı karakter belirsizlikleri, giriş dizelerinin yapısını analiz ederek çözülebilir, bu da tokenizasyonun ilk aşamasında kullanılmaktadır. (Grefenstette ve Tapanainen, 1994).

Rahman vd.'ne (2024) göre alt kelime tokenizasyon teknikleri, örneğin Byte-Pair Encoding (BPE) ve WordPiece, modern büyük dil modellerinin benimsediği alt kelime düzeyindeki tokenizasyon yaklaşımında önemli bir rol oynamıştır. Ancak mevcut tokenizasyon metodolojileri, dil bilgisel olarak anlamlı birimleri yakalamada temel zorluklarla karşılaşmaktadır; çünkü bunlar, dil bilimcilerin tanıdığı anlamın en küçük birimleri olan morfemlerle örtüşmeyebilir. Bu dilsel uyumsuzluğun sonuçları oldukça geniş kapsamlıdır. Sonuç olarak dilin doğru ve etkili bir biçimde anlaşılmasını zorlaştırabilir ve çeşitli dilsel özelliklerin kaybolmasına neden olabilmektedir.

Rahman vd.'ne (2024) göre önceleri doğal dil işleme (NLP) sistemlerinde, tokenizasyon genellikle boşluklara dayalıydı ve kelimeler arasındaki ayırıcılar boşluklarla

belirleniyordu. Bu basit yaklaşım, kelimeler arasında boşluk kullanmayan diller veya karmaşık kelime biçimlerine sahip diller (örneğin: ileri zaman sistemleri, eklemeli diller, ünlü uyumu vb.) ile çalışırken sınırlamalara sahipti. Örneğin, yaygın olarak kullanılan bir doğal dil işleme kütüphanesi olan NLTK'nın varsayılan tokenizatörü yalnızca boşluklara dayanarak tokenizasyon yapar ve anlamsal olarak anlamlı kelime alt birimleri hakkında hiçbir bilgi sağlamaz. Tokenizasyon temel olarak ikiye ayrılır.

3.1.1 Subword Tokenizasyonu

3.1.1.1 WordPiece tokenization

Rahman vd.'ne (2024) göre erken dönem alt kelime tokenizasyon yöntemlerinden biridir. Kelimeleri daha küçük birimlere ayırır, böylece modelin, kelime dağarcığında bulunmayan kelimeleri daha küçük alt kelime birimlerini birleştirerek temsil etmesini sağlamaktadır. Ancak, bu yöntem kelimeleri her zaman anlamsal olarak anlamlı alt birimlere ayırmaz.

WordPiece tokenizasyonu, bazen kelimeleri anlamsal olarak tam yansıtmayan şekillerde bölebilmektedir. Örneğin, WordPiece "mutsuzluk" kelimesini "mut," "suz" ve "luk" gibi parçalara ayırabilir. "Mut-" anlamlı bir kök ve "-luk" anlamlı bir ek olsa da, "suz" tek başına tam bir anlam ifade etmeyebilmektedir.

3.1.1.2 Byte-Pair encoding (BPE)

Başlangıçta veri sıkıştırma için geliştirilen bu yöntem modern Doğal Dil İşleme modellerinde yaygın olarak benimsenen bir alt kelime tokenizasyon yöntemi haline gelmiştir. Byte-Pair Encoding (BPE), metni karakter düzeyindeki desenlere dayalı olarak alt kelime birimlerine böler. Bu süreç, dinamik olarak bir alt kelime tokenleri sözlüğü oluşturur, böylece modelin sözlük dışı kelimeleri ele almasını ve morfolojik varyasyonları etkili bir şekilde temsil etmesini sağlamaktadır. Byte-Pair Encoding (BPE) ve WordPiece, NLP'de kullanılan alt kelime tokenizasyon yöntemleridir. BPE daha esnek olup alt kelime birimlerini karakter düzeyinde tanımlamaktadır. Bu sayede karmaşık dilleri ve sözlük dışı

kelimeleri daha iyi işleyebilmektedir. BPE, daha ince ayrıntılı temsiller sunar ve WordPiece'e kıyasla uygulanması daha kolaydır. BPE, dil bağımsızdır ve daha küçük sözlük boyutlarına ulaşabilir. BPE esnekliği, temsilleri ve dil desteği nedeniyle genellikle tercih edilmektedir (Rahman vd., 2024).

3.1.2 Diğer Tokenizasyon Metotları

3.1.2.1 SentencePiece tokenizasyonu

SentencePiece, kullanıcı dostu bir ara yüz sunarak tokenizasyon sürecini kolaylaştırmaktadır. Esnek segmentasyon teknikleri ile karakter bazında işlem yaparken, WordPiece belirli bir algoritma ile çalışır ve kelime bazında tokenizasyon yapmaktadır. SentencePiece, çok dilli doğal dil işleme uygulamaları için tasarlandığı için metin normalizasyonunu da desteklemektedir. WordPiece ise nadir kelimeleri çevirmek için geliştirilmiş bir yöntemdir ve normalizasyon süreci gerektirmeyen bir yaklaşım sunmaktadır (Rahman vd., 2024).

3.1.2.2 Unigram language model (ULM)

N-gram dil modelleri, kelime dizelerinin olasılıklarını hesaplamak için yalnızca önceki kelimelere bağlı kelime olasılıklarını kullanmaktadır. Unigram modeli, kelimeleri alt kelime birimlerine ayırarak daha esnek bir tokenizasyon sağlayarak dilin karmaşık yapılarının daha iyi temsil edilmesine ve modelin daha etkili çalışmasına olanak tanımaktadır (Arısoy vd., 2012).

3.1.2.3 Subword regularization

Bu metot, Kudo (2018) tarafından önerilen ve girdi metninde daha fazla çeşitlilik oluşturmak amacıyla olasılık hesaplarını kullanarak, bir alt birim segmentasyonu modelini uygulayan bir tekniktir. Bu çeşitlilik, modelin kelime yapısını daha iyi anlamasına yardımcı olarak genelleme yeteneğini artırmaktadır. Gizli değişken, kelimenin hangi alt birimlere bölüneceği ile ilişkili olmakta ve her eğitim aşamasında bu bölünme örnekleme yoluyla rastgele belirlenmektedir. Gizli değişken doğrudan hesaba katılamasa da, alt birim

düzenlemesi örnekleme yoluyla bunu yaklaşık olarak yapar. Örneğin kitaplar kelimesi her eğitim adımında farklı alt birime bölünebilir:

- Eğitim adımında: "kitaplar" ,"kitap", "lar"
- Başka bir eğitim adımında: "kit", "aplar"

Özetle tokenizasyon, metni daha küçük birimlere ayırma işlemidir ve birçok NLP uygulaması için temel bir ön işleme adımıdır. Modern NLP modelleri, kelimeleri alt kelime birimlerine ayırarak daha iyi genelleme yeteneği sağlamaktadır. Alt kelime tokenizasyonu, kelimeler ile karakterler arasında bir orta nokta oluşturur ve morfemlerle ilişkili anlamları korumaktadır. Örneğin, Byte-Pair Encoding (BPE) ve WordPiece, bu alt kelime tokenizasyonu tekniklerindendir. BPE, metni karakter düzeyindeki desenlere dayalı olarak bölerken, WordPiece nadir kelimeleri parçalamaktadır. Bununla birlikte, kelimeler her zaman anlamlı alt birimlere ayrılmaz. SentencePiece ve Unigram Dil Modeli (ULM), alt kelime tokenizasyonun da farklı yaklaşımlar sunmaktadır. SentencePiece, çok dilli işlemleri desteklemekte ve metin normalizasyonu yapmaktadır. ULM ise kelimeleri daha esnek alt birimlere ayırarak dilin yapısal özelliklerini daha iyi temsil eder. Subword Regularization, olasılıksal bir segmentasyon tekniğidir ve modelin kelimeleri daha iyi anlaması için kelime yapısını her eğitim adımında rastgele bölmektedir. Bu yöntem, kelimeleri gizli değişkenler olarak ele alır ve örnekleme yoluyla segmentasyon sağlamaktadır.

3.2 Word Embeddings

Lai vd.'ne (2016) göre word embeddings, dağıtık kelime temsili olarak da bilinmektedir. Bu temsil büyük bir etiketlenmemiş veri kümesinden kelimelerin hem anlamsal hem de sözdizimsel bilgilerini yakalayabilmektedir.

Harris'e (1954) göre dilin dağılımsal bir yapıya sahip olup olmadığına dair tartışmalarda, yapı terimi aşağıdaki şekilde kullanılmaktadır: Bir dizi fonem veya veri, belirli bir özellik açısından yapılandırıldığında, o özelliğe dayalı olarak setin üyelerini ve

aralarındaki ilişkileri tanımlayan bir sistem oluşturulabilmektedir. Bu anlamda, dil çeşitli bağımsız özelliklere göre yapılandırılabilmektedir. Dilin düzenli tarihsel değişim, sosyal etkileşim, anlam veya dağılım gibi unsurlar açısından ne ölçüde yapılandığı araştırmalarla belirlenebilmektedir. Bu bağlamda, her dilin dağılımsal bir yapı ile tanımlanabileceği, yani parçaların (nihayetinde seslerin) diğer parçalarla olan ilişkisi açısından açıklanabileceği tartışma konusu olabilmektedir. Dağılım, bir unsurun tüm çevrelerinin toplamı olarak anlaşılmaktadır. Bir unsurun çevresi, onun mevcut eşleşen unsurlarının dizisidir ve A'nın belirli bir pozisyondaki eşleşenleri, o pozisyondaki seçimi olarak adlandırılmaktadır.

Harris (1954) dağılımsal gerçeklerin yapı olanakları bölümünde, dilin yapısına dair bazı temel noktalar öne çıktığını belirtmektedir:

- **Dil Öğelerinin Dağılımı:** Dilin unsurları birbirine rastgele dağılmamaktadır; her bir eleman, belirli diğer unsurlara karşı belirli pozisyonlarda yer almaktadır. İnsanlar, konuşma sırasında istediği anlamı taşıyan kelimeleri bir araya getirdiğini düşünürken, aslında bu, düzenli olarak bir araya gelen sınıflardan seçilen üyelerin belirli bir sırayla birleştirilmesidir.
- **Sınıfların Dağılımı:** Sınıfların dağılımı, tüm kullanımları boyunca devam eden kısıtlamalara sahiptir; bu kısıtlamalar keyfi bir şekilde göz ardı edilmemektedir. Bazı mantıkçılar, doğal dillerin kesin bir dağılımsal tanımının imkânsız olduğunu öne sürse de, dildeki tüm unsurlar belirli sınıflara gruplandırılabilir ve bu grupların göreli dağılımları kesin bir şekilde ifade edilebilir.
- **Her Unsurun Diğer Unsurlara Göre Dağılımı:** Herhangi bir unsurun, diğer bir unsurla olan ilişkisi de, önceden belirlenmiş ölçütlere göre kesin bir şekilde ifade edilebilir. Bu sayede, dilin yapısı ve dağılımsal özellikleri hakkında daha derin bir anlayış geliştirmek mümkündür.
- Dördüncü olarak, **her bir unsurun göreli dağılımına dair kısıtlamalar**, birbirleriyle ilişkili ifadelerin bir ağı olarak en basit şekilde tanımlanabilir. Bu ifadelerden

bazıları, belirli diğer ifadelerin sonuçları cinsinden formüle edilir; bu da her bir unsurun toplam kısıtlamasının basit bir ölçümü yerine geçmektedir.

Bu olanakları Harris (1954) dilin matematiksel olarak ifade edilebileceği fikrine altyapı oluşturarak dilin yapısı olduğunu, bunun bilimsel bir çalışma olarak dilin tanımladığı verilerdeki ilişkiler ağını ifade ettiğini ve bu ilişkiler incelenen verilerde gerçekten mevcut olduğunu ileri sürer. Harris (1954) bu ilişkilerin dağılımsal düzenlilikler açısından incelenebileceğini ifade ederek dağıtık kelime temsilinin temellerini atmıştır.

Doğal dilin otomatik işlenmesinde metnin evrensel bir temsilini arama çabası merkezi bir öneme sahiptir. Bu alandaki büyük çıkış, word2vec veya GloVe gibi önceden eğitilmiş metin gömme modellerinin geliştirilmesiyle gerçekleşmiştir. Geçmiş yıllarda, denetimli modeller genellikle denetimsiz modellerden daha iyi sonuçlar göstermiştir. Ancak, son yıllarda öğretmensiz öğrenmeye dayalı modeller özellikle özel etiketli bir veri kümesi hazırlama zorunluluğu olmaması nedeniyle daha yaygın hale gelmiştir. Bu modeller, zaten mevcut olan veya otomatik olarak oluşturulan büyük metin koleksiyonları üzerinde öğrenim yapabilir ve bu nedenle çok daha büyük bir örnek üzerinde öğrenim yaparak derin öğrenim avantajlarından tam olarak yararlanabilirler (Koroteev, 2021).

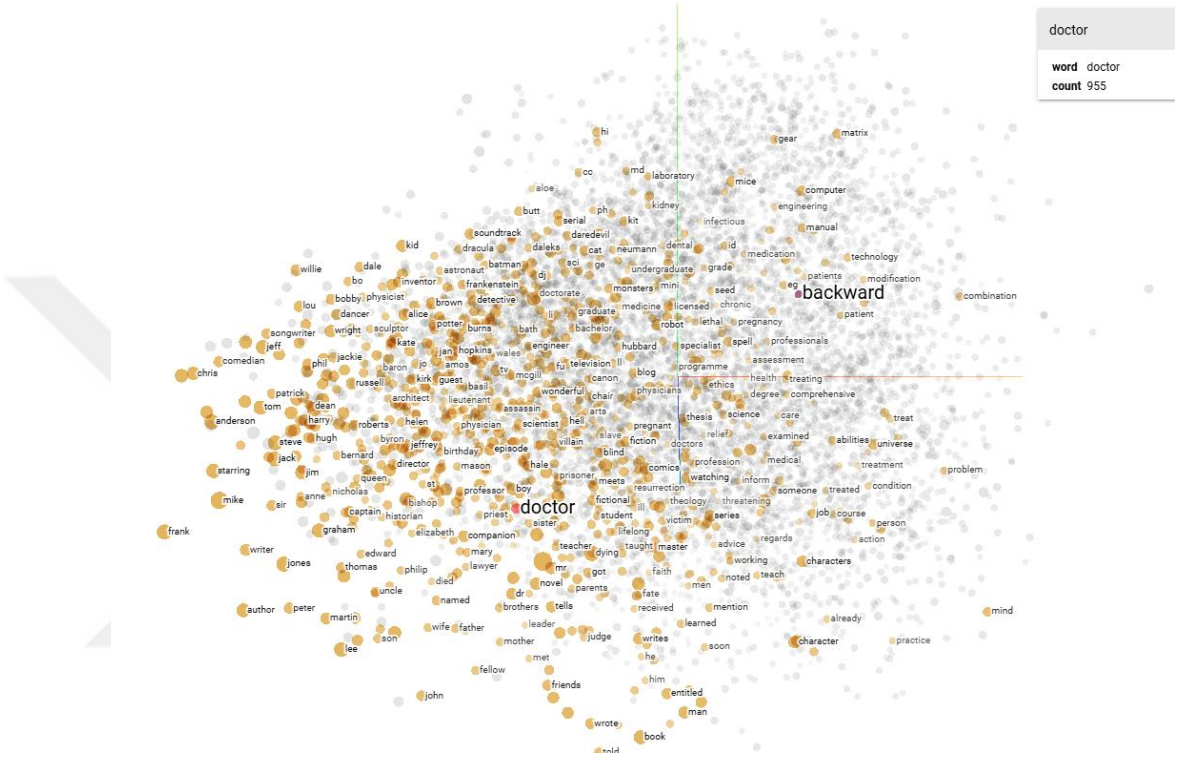
Metin verileri, dijitalleşmenin etkisiyle günümüzde hızla artmaktadır. İnternet, her gün milyonlarca belgeyle dolup taşarken, metin işleme görevleri insan için oldukça karmaşık hale gelmekte ve bu durum ne esnek ne de başarılı bir şekilde gerçekleştirilebilmektedir. Birçok makine öğrenimi algoritması, ham metni orijinal formatında yorumlayamaz; çünkü bu algoritmalar, herhangi bir görevi (örneğin, sınıflandırma veya regresyon) yerine getirmek için yalnızca sayılara ihtiyaç duyar. Bilgisayarların metni anlaması ve işlemesi için daha iyi bir temsil biçimine ihtiyaç duymaktadır. Kelime gömme (word embeddings), kelimelerin vektörler olarak kodlanması anlamına gelir ve son zamanlarda doğal dil işleme için bir özellik öğrenme tekniği olarak büyük ilgi görmektedir (Johnson vd., 2024).

Johnson vd.'ne göre (2024) Makinelerin metni işleyebilmesi için, kelime dağarcığındaki serbest biçimli metin terimleri sayısal değerlere (vektörlere) dönüştürülmelidir. Farklı makine öğrenimi teknikleri, metin verilerini otomatikleştirilmiş süreçler, tahmine dayalı modelleme, bilgi keşfi ve anlama amacıyla kullanmak için çeşitli işletmeler tarafından yaygın olarak kullanılmaktadır. Önemli sorunlardan biri, serbest biçimli metni makine öğrenimi algoritmalarının verimli bir şekilde kullanabileceği bir temsile dönüştürmektir. Bu dönüşümlerden biri one-hot encoding (tek sıcak kodlama) yöntemidir. Bu yöntem, kelime dağarcığındaki her kelimeyi, V farklı elemandan oluşan bir vektörde benzersiz bir indeksle dönüştürür. Sonuç olarak, bir kelime, uygun pozisyondaki bir (1) hariç, sıfırlarla (0) belirtilen bir vektörle temsil edilmektedir. Boyutun derecesi, kaynak kelime dağarcığındaki kelime sayısıyla belirlenmektedir. Bu yöntemde vektörün boyutu, kelime dağarcığındaki toplam kelime sayısına eşittir. One-hot encoding tekniğinin basitliğine karşın, belirli bir kelimenin anlamını ve kelimeler arasındaki farkı iletememektedir. Günümüz senaryosunun geniş veri işleme gereksinimleri nedeniyle, one-hot encoding, verileri binlerce boyutta temsil etmek için devasa bellek gereksinimleri nedeniyle uygulanamaz hale gelmiştir. Ayrıca, bu teknik kelime dağarcığındaki kelimeler arasında herhangi bir gizli ilişkiyi de barındırmamaktadır. Bu eksikliği aşmak için farklı kelime temsil yöntemlerine ihtiyaç vardır.

Kelime gömme (word embedding) yöntemi, daha önce bahsedilen geleneksel gömme tekniğinden farklılık göstermektedir. One-hot kodlanmış vektörler, kelime gömme yöntemleri kullanılarak yoğun temsillere haritalanmaktadır. Kelime gömme yaklaşımları, dil verilerinin anlamını yakalarken, genellikle kelime dağarcığı boyutundan daha düşük bir boyuta sahiptir. Daha yoğun kelime temsilleri elde etmenin, daha iyi kelime tahmini yapma imkânı sağladığı teorisi vardır. Örneğin, kelimelerin n-boyutlu vektör alanında "Doktor," "Domates" ve "Hekim" dağıtılmıştır. "Doktor" ve "Hekim" arasındaki mesafe, "Doktor" ile "Domates" arasındaki mesafeden daha azdır ve bu da

"Doktor" ile "Hekim" kelimelerinin benzer olduğunu göstermektedir. Bu teknik, kelimeler arasında gizli bir anlam bağlantısı kurar (Johnson vd., 2024).

Word embedding, kelimelerin matematiksel temsillerini oluştururken, kelimeler arasındaki bağlamsal ilişkileri dikkate almaktadır. Örneğin, "doktor" kelimesi, sağlık, tedavi, hastalık gibi terimlerle güçlü bağlara sahiptir. Şekil 3.1'den de görüleceği üzere word embedding algoritmaları, bu kelimelerin semantik olarak birbirine yakın olduğunu ve benzer bağlamlarda sıklıkla kullanıldıklarını gösteren vektörler üretir. Bu sayede, "doktor" kelimesi ve onunla ilişkilendirilebilecek kelimeler (örneğin, "hemşire", "hastane", "muayene") benzer vektörlere sahip olur, çünkü bu kelimeler genellikle aynı temada ve anlamda kullanılır. Bu tür bir temsil, makinelerin dilin derinliklerine inmelerine ve dilin anlamını daha iyi kavrayarak görevleri daha etkili bir şekilde yerine getirmelerine olanak tanımaktadır.



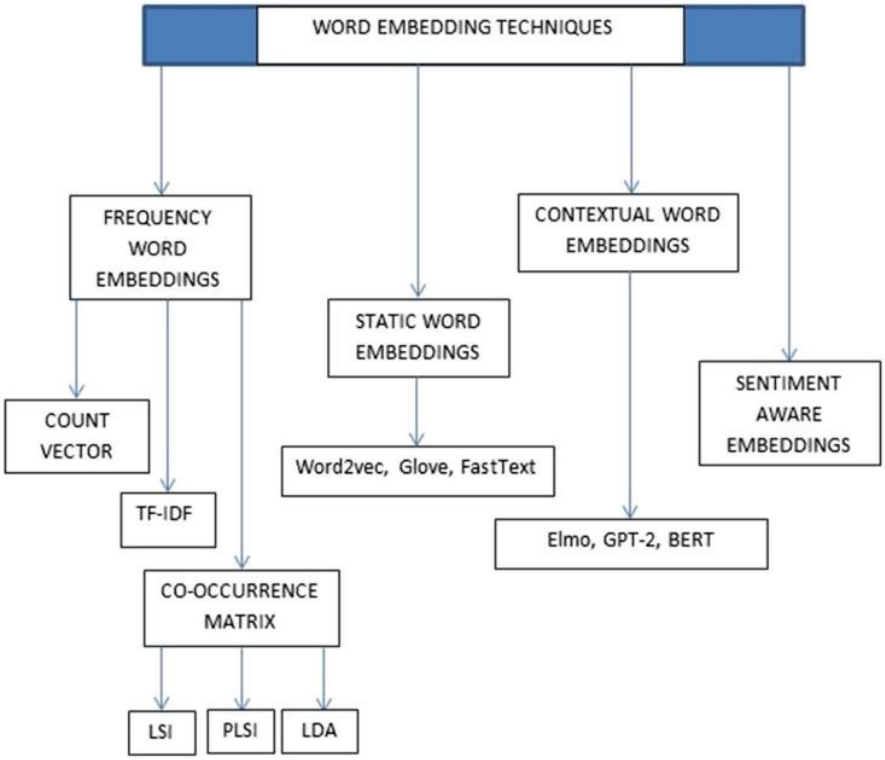
Şekil 3.1: Embedding Projeksiyonu

Kaynak: projector.tensorflow.org, 2024

McDonald ve Ramscar' a (2001) göre dağılım teorisi, benzer veya ilişkili anlamlara sahip kelimelerin aynı bağlamda meydana geldiğini öne sürmektedir. Sonuç olarak, sözdizimsel veya anlamsal olarak bağlantılı kelimler için gömme temsilleri, ilişkili olmayan terimlere kıyasla vektör alanında daha yakın yer almaktadır. Bu ilişki, bu gömme temsillerinin üretildiği metin verilerine veya külliyata tamamen bağımlı olmaktadır.

Johnson vd.'ne göre (2024) frekans tabanlı kelime gömme, statik kelime gömme, bağlamsal kelime gömme ve duygu farkındalıklı gömme yöntemleri, kelime gömme için kullanılan çeşitli tekniklerdir. Sayı vektörü, TF-IDF ve eş-oluşum matrisi, frekans tabanlı kelime gömme yöntemlerine örnek olarak verilebilmektedir. Eş-oluşum yöntemleri, Latent Semantic Indexing (LSI), Probabilistic LSI (PLSI) ve Latent Dirichlet Allocation (LDA) gibi tekniklerle daha da ayrıntılandırılabilinir. Statik kelime gömme

yaklaşımlarına örnek olarak word2vec, Glove ve FastText gösterilebilirken, bağlamsal kelime gömme yöntemleri arasında Elmo, GPT-2 ve BERT yer almaktadır. Bağlamsal gömme yöntemlerinin iyileştirilmesi, duygu farkındalıklı (sentiment-aware) teknikler ile sağlanmaktadır. Bu teknikler Şekil 3.2 'de görselleştirilmiştir.



Şekil 3.2: Farklı Kelime Gömme Tekniklerinin Şematik Temsili

Kaynak: Johnson vd., 2024

3.2.1 Frekans Tabanlı Word Embeddings

Johnson vd.'ne (2024) göre frekans tabanlı kelime gömme yöntemleri, belgede geçen kelimelerin sıklığına dayanmaktadır. Bu teknikler, benzersiz terimlerin önemini belirler ve oluşumlarını hesaplamaktadır. Gömme, metinlerden vektörler oluşturma işlemi olarak bilinir. BoW (Bag of Words) ve TF-IDF, metni sayısal vektörlere dönüştürmek için

yaygın olarak kullanılan yöntemlerdir. BoW yöntemi, her cümleyi kelime vektörleriyle temsil eder. Kelime dağarcığı, benzersiz kelimelerden oluşur ve vektör boyutu bu kelimelerin sayısına eşittir. Ancak bu yöntem, büyük belgeler için yüksek boyutlu ve seyrek vektörler üretir ve eş anlamlı kelimeleri yakalayamaz. TF-IDF (Term Frequency-Inverse Document Frequency), kelime sıklığına ve belgelerdeki dağılımına göre kelimelere ağırlık vermektedir. Terim Frekansı (Term Frequency), bir kelimenin belirli bir belgede ne kadar sıklıkla geçtiğini göstermektedir. Terim frekansı, Ters belge frekansı (IDF) ve TF_IDF hesabı aşağıdaki gibi yapılmaktadır:

$$TF_{t,\,d} = n_{t,d} / N_{t,d}$$

- $n_{t,\,d}$: "t" teriminin "d" belgesindeki görünüm sayısı.
- $N_{t,d}$: Belgedeki toplam terim sayısı.

Örnek olarak bir belgede “güzel” kelimesi 3 kez geçiyorsa ve toplam terim sayısı 100 ise bu durumda “güzel” kelimesinin terim frekansı (TF) 0.03 olarak hesaplanır.

$$IDF_{\,t} = \mathrm{Log}\,(N/n_t)$$

- N: Tüm belgelerin sayısı.
- N_t: Terimin bulunduğu belgelerin sayısı.

Örnek olarak 100 belgede “güzel” terimi 10 belgede geçiyorsa formüle göre IDF’i 1 olarak hesaplanır.

$$\text{TF-IDF} = \text{TF} \times \text{IDF}$$

Bu durumda “güzel” kelimesinin TF-IDF değeri 0.03 olarak hesaplanır.

Sonuç, terim-belge matrisi X elde edilir olarak (yukarıdaki örnekte 0.03) ve bu matrisin sütunları, belgeler için TF-IDF değerlerini içerir. TF-IDF puanı yüksek olan kelimeler daha önemli olarak kabul edilirken, düşük puanlı olanlar daha az kritik olarak görülmektedir. Bir gözlem olarak, TF-IDF yöntemi, farklı uzunluklardaki belgeleri sabit

uzunlukta sayısal listelere indirgememize yardımcı olmaktadır. Basitliği ve verimliliğine rağmen, TF-IDF, terimler arasındaki bağlantıların sadece küçük bir kısmını ortaya çıkarmaktadır; bu nedenle, eş anlamlılık ve çok anlamlılık gibi temel dilbilimsel fikirleri yakalayamamaktadır. Ayrıca, açıklama uzunluğunda yalnızca küçük bir azalma sağlamaktadır. Bu sorunları hafifletmek için araştırmacılar, LSI, PLSI ve LDA gibi çeşitli boyut indirgeme teknikleri geliştirmişlerdir (Johnson vd., 2024).

3.2.2 Matris ayrıştırma yöntemleri

Johnson vd.'ne (2024) göre Matris ayrıştırmanın amacı, yüksek boyutlu bir giriş matrisindeki en alakalı bilgileri, her biri tahmini veya doğru giriş yeniden yapılandırmasına sahip birkaç düşük boyutlu faktör matrisine sıkıştırmaktır. Ayrıştırma teknikleri, yüksek boyutlu ve seyrek veri yapılarına sahip ortamlarda seyrekliğin, toplam özellik sayısının ve gürültünün azaltılmasında kullanışlıdır. Bu teknikler ayrıca verilerin gizli özelliklerini de ortaya çıkarabilmektedir.

3.2.2.1 Latent semantik indeksleme (LSI)

Bu yöntem Latent Semantik Analiz (LSA) olarak da adlandırılmaktadır. Bu teknik, TF-IDF'in yaşadığı sorunları aşmayı amaçlar. Tekil Değer Ayrışımı (SVD) TF-IDF'e uygulandığında, çoğu varyansı koruyan örtük semantik bir alan oluşturulur (Golup ve Reinsch, 1971). Yeni uzaydaki her bir özellik, gerçek TF-IDF özelliklerinin doğrusal bir birleşimidir ve bu durum eş anlamlılık sorununu çözer. Bu tür bir yaklaşım, büyük bir derlemede önemli bir sıkıştırma sağlamaktadır. LSI, belirli bir belge kümesini analiz ederek birlikte ortaya çıkan istatistiksel kelime eşleşmelerini tespit ederek ve bu kelimeler ile belgelerin konularına dair fikir vermektedir (Johnson vd., 2024).

LSI'nin çözdüğü iki büyük sorun eş anlamlılık (synonymy) ve çok anlamlılıktır (polysemy). Eş anlamlılık, aynı nesnenin birden fazla şekilde ifade edilebilmesi anlamına gelmektedir. Örneğin, "2021'in En Güzel Kadını" ile "2021'in En Çarpıcı Kadını" gibi aramalar benzerlik göstermektedir. Çok anlamlılık ise çoğu kelimenin birden fazla anlama

sahip olabilmesi durumunu ifade etmektedir. Örneğin, "bank" kelimesi, farklı bağlamlarda farklı anlamlar taşımaktadır. Bir belgede "bank" kelimesi "nehir" kelimesi ile birlikte kullanıldığında, istatistiksel olarak "bank" kelimesinin bir nehir kıyısını ifade etmesi olasılık dâhilindedir. Bu tür bir teknik, verilerdeki gizli özellikleri ortaya çıkarmak ve gürültüyü azaltmak için terim-belge matrisini ayrıştırmaktadır (Johnson vd., 2024)

LSI hızlı ve verimli bir şekilde uygulanabilir, ancak birkaç büyük kısıtlamaya sahiptir. Önemli bir dezavantaj, oluşturulan semantik setin yeni belgeler eklendiğinde bu belgeleri işleyememesidir. SVD, boyutlar arasındaki tüm doğrusal ilişkileri kullanarak kelimenin geçtiği her metin örneğini tahmin eden vektörler üretir; bu yüzden herhangi bir hücredeki değeri değiştirmek, diğer tüm kelime vektörlerinin katsayılarını değiştirir. Bu yöntem, yeni içerik eklenirse tüm derlemenin yeniden indekslenmesini gerektirir ve dinamik olarak değişen derlemeler için sınırlı bir kullanım sunmaktadır (Johnson vd., 2024).

Özetle bu yöntem bir doküman koleksiyonundaki gizli anlamları keşfetmek için kullanılan istatistiksel bir model olarak öne çıkmaktadır. PLSI, kelimeler ile dokümanlar arasındaki ilişkileri olasılıksal bir çerçevede modelleyerek, gizli anlamsal yapıları ortaya çıkarmaya çalışmaktadır.

3.2.2.3 Olasılıksal latent semantik indeksleme (PLSI)

LSI'nin ifade gücünü artırmak için önerilmiştir. Bu teknik, yaygın olarak kullanılan başka bir belge modelidir. Gözlemlenmemiş bir konu “z” verildiğinde, PLSI modeli, belge etiketi “d” ve bir kelime w_n'nin koşullu olarak bağımsız olduğunu öne sürer. PLSI, Tekil Değer Ayrışımı (SVD) yerine olasılıksal bir yöntem kullanarak sorunu çözmektedir. Her PLSI belgesi birçok konuyu içermekte ve her kelime bu konulardan birine atanmaktadır. PLSI, LSI'ye olasılıksal bir konu ve kelime dağılımı yaklaşımı eklemektedir. Aslında LDA, PLSI'nin Bayes versiyonudur. Belge-konu ve kelime-konu dağılımları için Dirichlet önceliklerinin kullanılması, özellikle genelleştirmeyi artırmaktadır. LDA kullanılarak, her

biri en olası ilişkilendirildiği kelimelerle tanımlanan insan tarafından yorumlanabilmekte ve konular bir belge derlemesinden çıkarılabilmektedir (Johnson vd., 2024).

PLSI, daha esnek bir model olmasına rağmen, bazı sorunları da beraberinde getirmektedir. İlgili konuları temsil edecek olasılıksal modellerin eksikliği bu sorunlardan biri olarak görülmektedir. Ayrıca, PLSI parametrelerinin sayısı belgelerin sayısı ile doğru orantılı olarak artar ve bu da aşırı öğrenmeye yol açabilmektedir (Johnson vd., 2024).

3.2.2.4 Latent dirichlet allocation (LDA)

PLSI'nin aksine, her konu kelimeler üzerinde olasılıksal bir dağılımdır. LDA, aslında PLSI'nin Bayes versiyonudur. Belge-konu ve kelime-konu dağılımları için belge içindeki konuların dağılımını ve bir konu içindeki kelimelerin dağılımını önceden belirlemek için kullanılması, özellikle genelleştirmeyi artırır. LDA kullanılarak, insan tarafından yorumlanabilir konular bir belge derlemesinden çıkarılabilir ve her konu, en olası ilişkilendirildiği kelimelerle tanımlanmaktadır (Johnson vd., 2024).

3.3 Static Word Embedding

Statik kelime gömme fonksiyonu, her kelimeyi bir vektöre dönüştürmektedir. Bu vektörler, kelime hazinesinin boyutuna kıyasla yoğun (dense) olup daha düşük bir boyutluluğa sahip olmaktadır. Bu gömmeler, belirli bir boyutta sabit bir kelime hazinesi varsaydığından statiktir (Johnson vd., 2024).

Başka bir araştırmacı grubu, kelime gömme oluşturmak için sinir ağlarını kullanmaktadır. Sinir ağı yaklaşımları öngörücü olup, belirli bir girdi verisi için kabul edilebilir bir çıktıyı tahmin etmektedir. Word2vec iki katmanlı ileri beslemeli sinir ağı mimarisi kullanan çok iyi bilinen bir modeldir. Word2vec modelinin öğrendiği kelimelerin vektör temsili, anlamsal anlamlar taşır ve çeşitli NLP görevlerinde hayati bir rol oynamaktadır (Johnson vd., 2024).

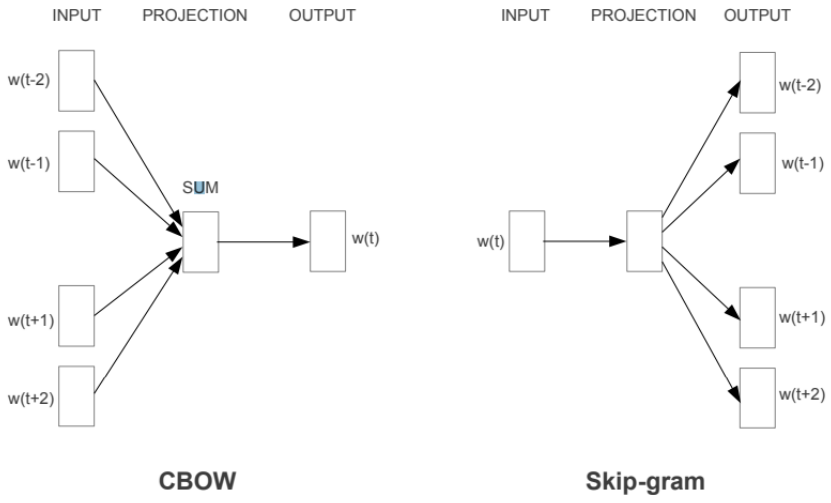
word2vec, iki şekilde uygulanmaktadır: (i) sürekli kelime torbası (CBOW) ve (ii) skip-gram modelleri olarak. CBOW modeli, hedef kelimeyi bağlam kelimelerine dayalı

olarak tahmin ederken, skip-gram modeli hedef kelimeye dayanarak bağlam kelimelerini tahmin etmektedir. Hem CBOW hem de skip-gram, çeşitli doğal dil işleme zorluklarını çözmek için kullanılabilecek yoğun ve düşük boyutlu kelime vektör temsilleri üretmektedir (Johnson vd., 2024).

Bu yaklaşımlar, önemli bir dezavantaja sahiptir: yalnızca yerel kelime eş-oluşum bilgisine sahip olup, küresel eş-oluşum bilgisi içermemektedir (Johnson vd., 2024).

3.3.1 Word2vec embeddings

Word2vec modelinin uygulanabileceği iki yöntem Şekil 3.3 de gösterildiği gibi, sürekli kelime torbası (CBOW) modeli ve skip-gram modelleri olarak belirtilmektedir.



Şekil 3.3: CBOW Skip-Gram Algoritmları

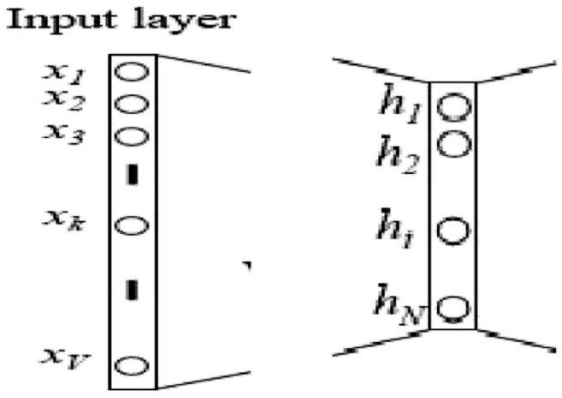
Kaynak: Mikolov vd.,2013

Bu modeller aşağıda detaylı biçimde ele alınmıştır.

3.3.2 Continious bag-of-words model

3.3.2.1 One-word context

Bu algoritmada, verilen bir bağlamda yalnızca bir kelimenin dikkate alındığı varsayılmaktadır; yani, bir bağlam kelimesi verildiğinde, model bir hedef kelime tahmin etmektedir. Bu, iki kelimeden oluşan bir modelle benzerlik göstermektedir. word2vec, tek bir gizli katmana sahip tam bağlantılı bir sinir ağı kullanmaktadır. Gizli katmandaki nöronlar, lineer nöronlardır. Giriş katmanındaki nöron sayısı, eğitim amaçları için kullanılan kelime dağarcığındaki kelime sayısına eşittir. X gizli katmanın boyutu, N ile gösterilen kelime vektörlerinin boyutuna eşittir; bu da N'nin kelimelerimizi temsil etmek için kullanılan toplam boyut sayısını belirttiği anlamına gelir. Bu, rastgele bir değer olup, sinir ağı için bir hiperparametredir. Çıkış katmanındaki toplam nöron sayısı, giriş katmanındaki nöron sayısına eşittir (Şekil 3.4) (Johnson vd., 2024).



Şekil 3.4: Bir Kelime Bağlamlı CBOW'da Giriş Katamanı

Kaynak: Johnson vd., 2024

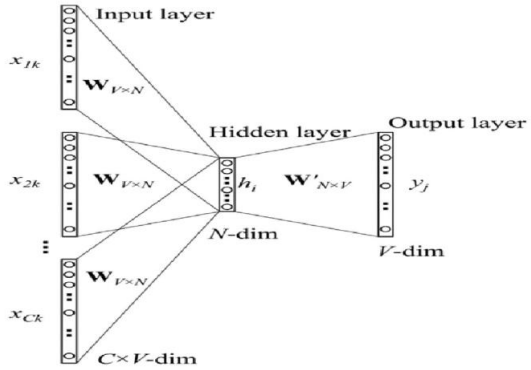
Şekil 3.4 de x değişkenleri giriş katmanına verilen kelimeleri temsil etmektedir. x_1 birinci kelime, x_2 ikinci kelime ve X_V v' inci kelimedir. V külliyat içerisindeki toplam kelime sayısıdır. H harfleri gizli katmandaki nöronları temsil etmektedir.

V külliyat içerisindeki kelime sayısı, yani giriş boyutu olarak tanımlanmaktadır. N gizli katmandaki nöron Sayısı, bu değer rastgeledir olarak atanmaktadır. Gizli katman boyutu, kelime vektörlerinin boyutunu belirlemektedir. $W_{V\times N}$ matrisini w_1 olarak gösterilecek olursa, ağırlık matrisi w_1, sinir ağının eğitildiğinde öğrendiği bilgi veya desenleri depolamaktadır. Ağırlık matrisi w_1, gömme katmanı olarak adlandırılmaktadır. Eğitim sırasında, giriş katmanının (giriş vektörü) ilk ağırlık matrisi (w_1) ile çarpımı gerçekleşmektedir. Başka bir ifadeyle, giriş vektörünün (one-hot kodlanmış vektör) ağırlık matrisiyle çarpılması durumunda, gizli katmanın değeri, ağırlık matrisindeki 1'e karşılık gelen değer olacaktır. Bir one-hot kodlanmış vektör ağırlık matrisi ile çarpıldığında, ağırlık matrisindeki yalnızca bir satır değerini koruyacak, diğer tüm satırlar ise 0 olacaktır. Sonuç olarak, kelime dağarcığındaki belirli bir kelime, örneğin n'inci kelime kullanıldığında, etki yalnızca ağırlık matrisindeki n'inci satırla sınırlandırılacaktır. Bunun nedeni, diğer tüm satırların 0 olmasından kaynaklanmaktadır. Sonuç olarak, ağırlık matrisinin n'inci satırı, kelime dağarcığındaki n'inci kelime hakkındaki tüm eğitilmiş bilgileri içermektedir. Eğitilmiş bir word2vec sinir ağından kelime vektörlerinin elde edilme şekli bu olacaktır. Böylece ağırlık matrisi N boyutlu kelime temsili ile V adet kelime korpusu için aşağıdaki gibi olacaktır (Johnson vd., 2024).

$$W_{V\times N} = \begin{bmatrix} w_{11} & w_{12} \cdots & w_{1N} \\ w_{21} & w_{22} \cdots & w_{2N} \\ \vdots & \vdots\vdots & \vdots \\ w_{V1} & w_{V2} \cdots & w_{VN} \end{bmatrix}$$

3.3.3 Multi-word context

Tek kelimeli bir bağlamda gizli katman çıktısını hesaplamak için, bağlam kelimesinin giriş vektörü doğrudan gizli katmana kopyalanır; ancak çok kelimeli bir bağlamda, girişteki bağlam kelimelerinin tüm vektörlerinin ortalaması alınır, bu ortalama gizli katmana aktarılan ağırlık matrisi ile çarpılır ve ortalama vektör çıktı olarak alınmaktadır.



Şekil 3.5: Birden Çok Kelime Bağlamlı CBOW'da Giriş Katamanı

Kaynak: Johnson vd., 2024

Bu metotta gizli katman çıktısı giriş vektörlerinin ortalaması alınarak ağırlık matrisi ile çarpılır. Şekil 3.5 de belirtildiği gibi, C giriş vektörlerinin ortalama değerini temsil etmektedir. Bu, bağlam kelimelerinin genel anlamını temsil eden tek bir vektör oluşturur ve böylece model, tahmin yapılacak hedef kelime için anlamlı bir temsil oluşturmaktadır (Johnson vd., 2024).

3.4 Contextual Metot

Son zamanlarda birçok araştırmacı, Elmo, Bert ve Xlnet gibi bağlamsal kelime gömme algoritmalarına odaklanmaktadır. Bu teknikler, bir kelimenin anlamının bağlam içinde sağlandığı varsayımına dayanmaktadır; bu nedenle, her bağlam için kelimenin benzersiz bir gömme temsili gerekmektedir. Çok anlamlılığı ele almak için, sayım tabanlı ve sinir ağı tabanlı yaklaşımlarda yüksek kaliteli kelime gömmeleri oluşturmak için ince ayar yapılabilmektedir (Johnson vd., 2024).

3.5 Text Classificaiton

Metin sınıflandırma (Text Classification - TC), büyük önem taşıyan bir görev olup, metin madenciliği ve doğal dil işleme (NLP) alanlarındaki son gelişmeler sayesinde giderek daha fazla ilgi görmektedir. TC yöntemlerinin ortak amacı, verilen bir metne

önceden tanımlanmış bir etiketi atamaktır; ancak, bu etiketleme farklı alanlarda özelleşmiş birçok yöntemi kapsayabilmektedir. TC'nin klasik örnekleri arasında bilgi erişimi, konu etiketleme, duygu analizi ve haber sınıflandırma yer almaktadır. Bununla birlikte, TC uygulamaları yalnızca basit sınıflandırma ile sınırlı kalmaz; özetleme ve soru-cevap sistemleri gibi daha karmaşık alanlara da yayılmaktadır. Bu durumda, "etiket" kavramı, bir cevabın veya özetlenecek cümlenin seçilmesi gibi daha geniş bir anlam kazanmaktadır. Metinsel bilginin günümüzdeki üretim hızı, bu görevlerin manuel çözümlerle yapılmasını neredeyse imkânsız hale getirmiştir; dolayısıyla, TC yöntemleri sadece faydalı değil, aynı zamanda zorunlu hale gelmektedir. Bu bağlamda, doğru ve tarafsız T.C. sistemlerinin geliştirilmesi büyük önem taşımaktadır (Gasparetto vd., 2022).

Gasparetto vd.'ne (2022) göre metin sınıflandırma görevleri aşağıdaki gibidir;

- Duygu analizi (Sentiment Analysis - SA): Bir metinde ifade edilen duygusal durumları ve öznel bilgileri anlamlandırma görevidir; genellikle duygusal tepkilere göre kategorize edilmektedir.

- Konu etiketleme (Topic Labelling - TL): Bir metnin içeriğindeki ana temaları veya konuları tanımlama görevi olarak bilinmektedir.

- Haber sınıflandırma (News Classification - NC): Haber metinlerini belirli kategorilere (örneğin konular veya kullanıcı ilgileri) göre sınıflandırma işi olarak tanımlanır.

- Soru-cevap (Question Answering - QA): Bir soruya verilen potansiyel aday cümlelerden doğru cevabı seçme görevidir; genellikle ikili veya çok sınıflı sınıflandırma olarak ele alınmaktadır.

- Doğal dil çıkarımı (Natural Language Inference - NLI): İki cümlenin birbiriyle ilişkili olup olmadığını belirleme görevidir; bu ilişki, cümlelerden birinin diğerini içerip içermediğini veya hiç ilişki olmadığını göstermektedir.

- Adlandırılmış varlık tanıma (Named Entity Recognition - NER): Yapılandırılmamış bir metinde yer alan özel isimleri bulma ve bunları önceden tanımlanmış kategorilere göre etiketleme olarak tanımlanır.

- Sözdizimsel çözümleme (Syntactic Parsing - SP): Sözcüklerin biçimsel ve sözdizimsel özelliklerini (örneğin, sözcük türü etiketleme, bağımlılık ilişkileri ve anlamsal rol etiketleme) tahmin eden görevler dizisi olarak bilinmektedir.

Derin öğrenme modellerinin ortaya çıkışı, metin sınıflandırma dâhil olmak üzere yapay zekânın tüm alanlarını etkilemiştir. Bu yöntemler, karmaşık özellikleri elle mühendislik yapmaya gerek kalmadan modelleyebildikleri için ilgi görmektedir; böylece belirli bir alan bilgisi gereksinimi de kısmen ortadan kalkmaktadır. Bunun yerine, metinsel birimlerin etkili temsillerini çıkartabilen sinir ağı mimarileri geliştirilmektedir. Son yıllardaki gelişmeler, anlamsal olarak anlamlı ve bağlamsal temsillerin ortaya çıkmasına katkıda bulunarak özellikle başarılı olmuştur. Otomatik özellik çıkarımı, metinsel veriyi modellemede oldukça avantajlıdır çünkü bir belgenin dilbilimsel yapısından yararlanabilmektedir. Bu yapı, dili anladığımızda bize sezgisel olarak anlam ifade ederken, makineler için genellikle anlaşılması zor olan bir özellik olarak ifade edilmektedir (Gasparetto vd., 2022).

3.6 Named Entity Recognition (NER)

Adlandırılmış Varlık Tanıma (NER), bir metin içinde gerçek dünya nesnelerini ifade eden alt dizileri tanımlamayı ve bunların türlerini belirlemeyi amaçlamaktadır. NER, Adlandırılmış Varlık Tanıma anlamına gelir ve doğal dil işleme (NLP) alanının bir alt görevi olup, metin içinde adlandırılmış varlıkları tanımlamaya ve sınıflandırmaya odaklanmaktadır. Adlandırılmış varlıklar, insanlar, organizasyonlar, yerler, tarihler, miktarlar, biyomedikal alandaki gen ve protein adları gibi gerçek dünyadaki nesnelere atıfta bulunan özel kelimeler veya ifadeler olarak tanımlanmaktadır. NER, bu varlıkları bulmayı ve bunları önceden tanımlanmış kategorilere ayırmayı amaçlamaktadır. Adlandırılmış Varlık Tanıma (NER), yapılandırılmamış metinlerden kişileri,

organizasyonları ve yerleri gibi varlıkları tanımlayarak bu varlıkları sınıflandıran önemli bir tekniktir. NER, bilgi çıkarımı, bilgi erişimi, belge özetleme, sosyal medya izleme, sanal asistanlar ve soru yanıtlama gibi birçok alanda kullanılmaktadır. NER, metinlerdeki varlıkları tanımlayarak arama sonuçlarını iyileştirebilmekte ve özetlerin kalitesini artırabilmektedir. Ayrıca, markaların sosyal medyadaki görünürlüğünü izleyebilir ve sanal asistanların kullanıcılara bağlama dayalı yanıtlar sunmasına yardımcı olabilmektedir. NER, belirsizliği giderme ve dil çevirisi süreçlerinde de kritik rol oynamaktadır, çünkü doğru bağlamla varlıkları ayırt ederek doğru sonuçlar sunmaktadır (Keraghel vd., 2024).

Keraghel vd.'ne (2024) göre Adlandırılmış Varlık Tanıma (NER) için kullanılan çeşitli metotlar vardır;

- İlk olarak, bilgi tabanlı yöntemler öne çıkmaktadır. Bu yöntemler, dilbilimsel kurallara ve sözlük kaynaklarına dayanarak varlıkları tanımayı amaçlamaktadır. Bu tür yöntemler, etiketlenmiş veriye ihtiyaç duymadan kurallara dayalı olarak varlıkları tanımlar ve bu kurallar genellikle uzmanlar tarafından hazırlanmaktadır.

- Bir diğer yaklaşım ise özellik mühendisliğine dayalı yöntemlerdir. Bu yöntemler, adlandırılmış varlıkları verilerden otomatik olarak çıkarmayı amaçlamakta ve genellikle üç grupta sınıflandırılmaktadır: (i) denetimsiz öğrenme, (ii) denetimli öğrenme ve (iii) yarı denetimli öğrenme. Denetimsiz öğrenme, veri içindeki benzerliklere dayalı olarak varlıkları gruplar ve etiketlenmemiş verileri kullanarak varlıkların ortak özelliklerini çıkarmaktadır. Yarı denetimli öğrenme, etiketli ve etiketlenmemiş verilerin birleşimini kullanarak model performansını artırmaya çalışılmaktadır. Bu tür yöntemler, etiketli verilerin az olduğu durumlarda faydalı olmaktadır. Denetimli öğrenme ise etiketli verilerle kurallar öğrenir ve bu öğrenilen kurallar yeni verilerde uygulanarak varlıkları sınıflandırmaktadır.

- Derin öğrenme tabanlı yaklaşımlar ise NER alanında son yıllarda önemli bir gelişim göstermiştir. Bu yöntemler genellikle CNN'ler (Convolutional Neural Networks)

ve RNN'ler (Recurrent Neural Networks) gibi ağlarla uygulanmaktadır. Bu tür yöntemler, NER görevini otomatikleştirir ve manuel özellik mühendisliğine olan ihtiyacı ortadan kaldırmaktadır. Derin öğrenme tabanlı yaklaşımlar, verilerin temsil edilmesinden bağlamın kodlanmasına ve varlıkların çözülmesine kadar üç aşamadan oluşmaktadır. Bu yöntemler, hem yerel hem de küresel bağlamı dikkate alarak varlıkları daha doğru şekilde tanımlamayı hedeflemektedir.

Sonuç olarak, NER için kullanılan yöntemler arasında bilgi tabanlı yaklaşımlardan derin öğrenmeye kadar geniş bir yelpazede farklı teknikler bulunmaktadır. Her bir yaklaşım, belirli veri türleri ve kullanım senaryolarına göre avantajlar sunar.

3.7 Sequence-to-Sequence Models

Sequence-to-sequence (seq2seq) modelleri, sırasal problemleri çözmek için yaygın bir çerçeve sunmaktadır (Sutskever, 2014, akt. Keneshloo vd., 2019). Bu modellerde, giriş bir dizi veri biriminden oluşur ve çıkış da bir veri birimi dizisi olarak ifade edilir. Geleneksel olarak, bu modeller, öğretici zorlaması (teacher forcing) adı verilen bir mekanizma kullanılarak, gerçek doğru (ground-truth) dizisiyle eğitilmektedir. Burada öğretici gerçek doğru dizisi olarak bilinir (Bengio vd. 2015, akt. Keneshloo vd., 2019).

Se2seg, giriş dizisini bir çıktı dizisinin ardışık bir sıralamasına dönüştüren etiket ve değer dikkate alınarak kodlayıcı-çözücü tabanlı makine yorumlamasıyla, tekrarlayan sinir ağları (RNN'ler) kullanılarak uygulanabilmektedir. Fikir, iki RNN'yi özel bir token ile işbirliği yaparak ve önceki sıralamadan bir sonraki durum sıralamasını tahmin etmeye çalışarak kullanmaktır. Sequence-to-Sequence modelleri ayrıca bağlantısal zaman sınıflandırması (CTC) ve dikkat tabanlı modeller olarak bilinen yöntemlerle de uygulanabilmektedir. Sequence-to-sequence modeli ilk olarak Google tarafından makine çevirisi için oluşturulmuş ve modeli tek bir komutla eğitmek için tanıtılmıştır. Model, ayrıca kolayca yeniden üretilebilir ve genişletilebilir olacak şekilde tasarlanmış ve kod dosyaları ölçülü bir şekilde organize edilmiştir, böylece genişletilmesi ve yeniden yaratılması kolay olmaktadır (Yousuf vd., 2021).

Sinir ağları, hayvanların beyinlerinden oluşan biyolojik sinir ağlarından ilham alarak tasarlanmış ve gelişim, ilerleme ve yüksek düzeyde kavrayış gerektiren karmaşık problemleri çözmek için tasarlanmıştır. Bu tür zor görevleri, örneğin konuşma tanıma ve makine çevirisi gibi görevleri oldukça iyi bir şekilde yerine getirebilen birçok sinir ağı türü vardır; bunlardan biri de tekrarlayan sinir ağlarıdır (Yousuf vd., 2021).

Yousuf vd.'ne (2021) göre sequence-to-sequence algoritma modellerini uygulamak için birkaç farklı yaklaşım bulunmaktadır. En yaygın kullanılan modeller arasında bağlantısal zaman sınıflandırması (CTC), tekrarlayan sinir ağları (RNN) ve dikkat tabanlı modeller yer almaktadır. Bağlantısal Zaman Sınıflandırması (CTC), giriş dizilerinin hedef etiketler olmadan uçtan uca işlenmesini sağlayan bir yöntemdir ve özellikle tekrarlayan sinir ağları (RNN) kullanarak çıkış koşulunun olasılığını tanımlamaktadır. CTC, etiketlerin tekrarını ve boş etiket kullanımını içeren özel bir yapı sayesinde, çıktı dizilerinin giriş dizilerinden daha kısa olduğu durumlarda sıralı-sıralı (sequence-to-sequence) modellerin verimli şekilde çalışmasına olanak tanımaktadır. Diğer yandan, RNN'ler, iki ağın birlikte çalışarak giriş dizisinden çıkış dizisini tahmin etmeye yönelik bir model oluşturmaktadır. Dikkat tabanlı modeller ise, belirli katmanlara daha fazla ağırlık vererek, geçmiş tahminlerden ve akustik verilere dayalı olarak çıktıyı şekillendiren bir çözücü kullanır, bu da modelin verimli ve doğru tahminler yapmasını sağlamaktadır. Bu yöntemler, özellikle makine çevirisi ve konuşma tanıma gibi uygulamalarda büyük avantajlar sunmaktadır.

Sonuç olarak, sequence-to-sequence (seq2seq) modelleri, özellikle dil işleme ve benzeri sırasal görevlerde etkili çözümler sunmaktadır. Bu modeller, giriş ve çıkış dizileri arasındaki dönüşümü sağlamak için çeşitli yöntemlerle uygulanabilir, bunlar arasında tekrarlayan sinir ağları (RNN), bağlantısal zaman sınıflandırması (CTC) ve dikkat tabanlı modeller öne çıkmaktadır. CTC, etiketlerin tekrarı ve boş etiketlerin kullanımıyla giriş ve çıkış dizileri arasındaki uyumsuzluğu giderirken, RNN'ler iki ağın işbirliğiyle tahmin yapması sağlanmaktadır. Dikkat tabanlı modeller ise, geçmiş bilgiden daha etkin

yararlanarak çıkış üretir. Bu yöntemler, makine çevirisi, konuşma tanıma gibi uygulamalarda önemli avantajlar sağlayarak, verimli ve doğru sonuçlar elde edilmesine olanak tanır.

3.8 Dependency Parsing

Bağımlılık çözümleme, doğal dil işleme alanında cümlelerdeki kelimeler arasındaki bağımlılık ilişkilerini analiz ederek dil bilgisel yapı hakkında bilgi elde etmeyi amaçlayan önemli bir tekniktir. Bu işlem, makine çevirisi, anlamsal çözümleme ve ilişki çıkarımı gibi çeşitli NLP görevlerinde kullanılır. Bağımlılık çözümleme modelleri genellikle denetimli öğrenme kullanılarak eğitilir, ancak bu yöntem, doğru çözümleme ağaçlarıyla etiketlenmiş eğitim verilerine ihtiyaç duyar ki bunlara "treebank" adı verilir. Ancak, bu tür veri kümeleri her dil veya yeni bir alan için mevcut değildir ve kaliteli bir treebank oluşturmak zaman alıcı ve maliyetlidir. Bu zorluklara rağmen, denetimsiz öğrenme, özellikle denetimsiz bağımlılık çözümlemesi (ya da bağımlılık dilbilgisi üretimi) önemli bir araştırma alanıdır. Denetimsiz öğrenme, etiketlenmiş cümleler olmadan bağımlılık çözümleyicilerini öğrenmeyi amaçlar ve bu yöntem, insan etiketlemesi gerektirmeyen büyük metin verilerinin nasıl kullanılabileceği konusunda yeni yollar sunmaktadır. Bu tür teknikler, diğer NLP görevleri ve dil edinimi çalışmaları için de faydalı olabilmektedir. Ayrıca, denetimsiz çözümleme yöntemleri, insan dilinin öğrenilmesi üzerine yapılan bilişsel araştırmalarla paralellik göstermekte ve bu alandaki çalışmaları desteklemektedir. Bağımlılık çözümleme, bir giriş cümlesi x için sözdizimsel bağımlılık ağacını z'yi keşfetmeyi amaçlamaktadır. Burada x, x_1,x_2…x_n, kelimelerinin sırasıdır ve n uzunluğuna sahiptir. Cümle başında genellikle bir "boş" kök kelimesi x_0 eklenir. Bağımlılık ağacı z, kök kelime x_0 etrafında yönlendirilmiş bir ağaç yapısı oluşturan kelimeler arasındaki yönlendirilmiş kenarlardan oluşur. Her bir kenar, bir anahtar kelimeden (baş kelime olarak da adlandırılır) bir alt kelimeye doğru işaret etmektedir (Han vd., 2020).

Han vd.'ne (2020) göre bağımlılık çözümlemenin ilgili olduğu alanlar denetimli, alanlar arası ve diller arası, denetimsiz anlam gruplaması ve aşağı görevlerde latent ağaç

modelleri olarak belirlenmiştir. Denetimli bağımlılık çözümleme, eğitim cümlelerinden elle etiketlenmiş bağımlılık çözümleme ağaçlarıyla bir çözümleyici eğitmeyi amaçlar ve genellikle grafik tabanlı ve geçiş tabanlı yaklaşımlar olarak iki ana kategoriye ayrılmaktadır. Grafik tabanlı yaklaşımlar (i), cümledeki kelimeler arasındaki tüm bağlantıları içeren bir grafik üzerinden en iyi ağacı arar, geçiş tabanlı yaklaşımlar (ii) ise ağacı soldan sağa doğru kademeli olarak inşa eder. Her iki yaklaşım da kaynak açısından zengin dillerde güçlü sonuçlar verirken, bu çözümler veri kümesi açısından kısıtlı alanlarda sınırlı kalmaktadır. Bu tür kaynak eksikliklerini aşmak için, kaynak dillerindeki çözümlenmiş modeller hedef dil veya alana uyarlanabilir, ancak denetimsiz çözümleme bu durumda daha zorlu bir yol sunar çünkü hedef alan için herhangi bir etiketlenmiş veriye de ihtiyaç duyulmamaktadır. Ayrıca, denetimsiz anlam gruplaması çözümlemesi, daha karmaşık olarak kabul edilse de, son yıllarda önemli bir araştırma konusu haline gelmiştir. Bu alanda yeni teknikler geliştirilmiş ve bu teknikler, alt görevlerin performansını iyileştirerek sonuca katkı sağlamayı hedefleyen latent (gizli) ağaç modellerinde de kullanılmıştır. Bu modeller, çözümleme ağacını bir latent değişken olarak ele alarak ve uçtan uca öğrenme süreçlerini optimize etmek için özel algoritmalar kullanmaktadır (Han vd., 2020).

Özetle, bağımlılık çözümleme yöntemleri, doğal dil işlemede dilin yapısal ve anlamsal analizini sağlama yönünde çeşitli avantajlar sunmaktadır. Denetimli yöntemler, veri açısından zengin dillerde güçlü performans gösterirken, özellikle kısıtlı kaynaklara sahip diller ve alanlar için veri eksikliği önemli bir engel teşkil etmektedir. Bu eksikliği gidermek için alanlar arası ve diller arası uyarlamalar önerilse de, denetimsiz öğrenme bu konuda daha zorlu ancak umut vaat eden bir çözüm olarak öne çıkmaktadır. Ayrıca, yeni geliştirilen gizli ağaç modelleri, çözümleme sürecini optimize etmeye yönelik yenilikçi yaklaşımlar sunmaktadır. Bu modeller, etiketlenmiş veri olmadan bağımlılık çözümleme yapılabilmesini sağlayarak doğal dil işleme alanında yeni araştırma fırsatları yaratır ve farklı NLP görevlerinde uygulanabilirliği artırmaktadır.

4.UYGULAMA

4.1 Giriş ve Genel Bakış

Son yıllarda, doğal dil işleme (NLP) alanındaki gelişmeler, dil modellerinin performansını önemli ölçüde artırmıştır. Özellikle önceden eğitilmiş dil modelleri, dilin yapısını anlamada ve metin işleme görevlerinde önemli başarılar elde etmiştir. Ancak, mevcut dil modellerinin çoğu, özellikle Türkçe gibi dil yapısı açısından zengin dillerde, yeterince optimize edilmemiştir. Bu uygulama, önceden eğitilmiş dil modellerinin Türkçe özelindeki performansını iyileştirmek için daha verimli bir alternatif yöntem önermeyi hedeflemektedir.

Çalışmanın amacı, Türkçe metinlerin özetlerini daha doğru ve verimli bir şekilde çıkarabilen bir model geliştirmektir. Özellikle, Türkçe bir paragrafın özetini çıkarma süreci üzerinde yoğunlaşılmakta olup, mevcut dil modellerinin bu görevi nasıl daha iyi gerçekleştirebileceği araştırılmaktadır. Farklı önceden eğitilmiş dil modellerinin özetleme performanslarının değerlendirilmesi, bu araştırmanın temel adımlarından biridir. Elde edilen sonuçlar, özetleme görevindeki başarıyı artıran yeni yöntemlerin geliştirilmesine katkı sağlamayı amaçlamaktadır.

Araştırma süreci, Python programlama dili kullanılarak gerçekleştirilmiş ve elde edilen sonuçlar, çeşitli performans metrikleriyle değerlendirilmiştir. Bu metrikler ROUGE skoru gibi ölçütlerdir ve modelin doğruluğunu ve etkinliğini ortaya koymak için kullanılmıştır. Bu çalışma, teorik gelişmelerin yanı sıra, pratikte daha iyi Türkçe özetler çıkarabilen modellerin üretimine katkı sağlamayı hedeflemektedir.

Türkçe özetleme konusunda yapılan bu araştırma, doğal dil işleme alanında önemli bir boşluğu doldurmayı amaçlamakta olup, Türkçe metinlerin daha iyi anlaşılmasını ve özetlenmesini sağlayacak yenilikçi çözümler geliştirmeyi hedeflemektedir.

4.2 Gereksinim Analizi

Bu uygulama, doğal dil işleme (NLP) alanında kullanılan önceden eğitilmiş modellerin performansını değerlendirmek amacıyla geliştirilmiştir. Uygulama, Google tarafından 770 milyon parametre ile eğitilmiş T5-Large (Raffel vd., 2020) modelini kullanarak Huggingface açık veri tabanı olan Wikipedia Türkçe veri setinden (Güngör, 2023) toplu olarak hazırlanmış Türkçe wikipedia verisinden rastgele metinlerin T5-Large modeli kullanılarak özetlerini oluşturacaktır. Yine aynı metinden kosinüs benzerlikleri en yakın yedi cümle seçilecek ve bu yedi cümle T5-Large modeliyle özetlenecektir. Elde edilen bu özetler, veri tabanında metinlerin özeti olarak var olan ve insan özeti kalitesinde olduğu doğrulanmış özetle değerlendirmek için ROUGE skorları ile performans metriklerini hesaplayacaktır. Sonuçlar iki aşamada ele alınacaktır: (i) Metindeki kosinüs benzerliği en yüksek yedi cümlenin T5-Large modeli ile elde edilen özetleri, yalnız T5-Large ile özetlenen ROUGE değerleri ile karşılaştırılarak analizi yapılacak, (ii) Uygulamada elde edilen tüm örneklerin ROUGE sonuçları ile t-testi yapılarak sonuç genelleştirilecektir.

4.2.1 Fonksiyonel Gereksinimler

4.2.1.1 Metin girişi

Uygulama, veri tabanından rastgele seçilmiş metin ve özetlerini giriş olarak almaktadır. Bu metinler, özetleme yapılacak metinlerdir ve uygulama bu metinleri alarak işleme başlayacaktır. Veri tabanına Python kodu ile bulut sisteminden erişilecektir. Bu metinler kontrol edilecek ve gerekli görüldüğü takdirde veri temizleme işlemi gerçekleştirilecektir.

4.2.1.2 Model ile özetleme

Yazılım, özetleme işlemini yalnızca T5-Large modelini kullanarak yapacaktır. T5-Large modeli, metni doğrudan özetler. Karşılaştırılacak özet, SentenceTransformer modeli ile kosinüs benzerliği en yakın yedi cümleyi örnek metinden alarak hesaplayacak

ve T5-Large modeline özetlemesi için gönderecektir. Kosinüs benzerliği için SentenceTransformer modeli kütüphanesi kullanılarak gerçekleştirilecektir. Başka bir ifade ile bu model metnin her cümlesi için vektörler oluşturmak ve bu vektörlerden benzer yedi cümleyi seçmek için kullanılacaktır.

4.2.2 Performans ölçümü

Özetleme işlemi gerçekleştikten sonra, uygulama özetlerin doğruluğunu ve geçerliliğini değerlendirir. Bu değerlendirme ROUGE skoru gibi metrikler kullanılarak yapılır. ROUGE-1, ROUGE-2, ROUGE-L gibi metrikler özetin kapsamını ve kalitesini ölçecektir.

4.2.3 Sonuçların Görüntülenmesi

Uygulama, özetleme işlemi ve performans değerlendirmesi sonuçlarını bir çalışma tablosu dosyasına kayıt edecek ve böylece değerlendirilmesi için veri tablosu sağlayacaktır. Bu sonuçlar, her bir model için ayrı ayrı ROUGE skorları olarak raporlanır. Bu raporlardaki veriler Pandas, MatPlotLib ve SeoBorn kütüphaneleri kullanılarak görselleştirilecek ve tezin sonuçları kısmında gösterilerek analizi yapılacaktır.

4.3 Teknik Gereksinimler

4.3.1 Yazılım dili ve çalışma Ortamı

Yazılım, Python programlama dili kullanılarak geliştirilmiştir. Python 3.9 sürümü, doğal dil işleme kütüphanelerine (Transformers, Sentence-Transformers, Scikit-learn, Rouge-Score) ve metrik hesaplamalarına destek sunan ideal bir ortam sağlar.

4.3.2 Dil Modelleri ve Ana Yazılım Kütüphaneleri

Yazılımda kullanılan dil modelleri:

T5-Large: Türkçe metinlerin özetlenmesi için kullanılır.

SentenceTransformer: Metinleri matematiksel olarak anlamsal vektörler ile ifade eden bir kütüphanedir. Her cümle transformer tabanlı modeller kullanılarak n boyutlu bir

vektör ile temsil edilir. Bu vektörler, çok boyutlu bir uzayda cümlelerin anlamını taşır ve cümleler arasındaki benzerlik genellikle kosinüs benzerliği formülü ile hesaplanır Model ve kütüphaneler Hugging Face Transformers ve Sentence-Transformers gibi önceden eğitilmiş modelleri sağlayan kütüphanelerle bütünleşmiştir.

4.3.3 Donanım Gereksinimleri

Yazılımın verimli çalışabilmesi için yeterli işlemci ve bellek kapasitesine ihtiyaç vardır. Kullanıcı, modelin büyüklüğüne göre, özellikle T5-Large için güçlü bir donanım (en az 128 GB RAM, çok çekirdekli işlemci) kullanmalıdır. GPU desteği, modelin hızlı çalışması için faydalı olmaktadır.

4.3.4 Performans Ölçümü ve Değerlendirme

Performans metriklerinin hesaplanmasında ROUGE metrikleri (ROUGE-1, ROUGE-2, ROUGE-L) kullanılarak özetin kapsamı, doğruluğu ve anlamlılığı ölçülmektedir.

ROUGE-1 (Recall-Oriented Understudy for Gisting Evaluation), metin özetleme sistemlerinin doğruluğunu değerlendirmek için kullanılan en temel metriklerden biri olarak kabul edilmektedir. ROUGE-1, referans özet ile model tarafından oluşturulan özet arasındaki birer kelimelik örtüşmeleri (unigram overlap) hesaplar. Bu metrik, özetin referans metni ne kadar doğru bir şekilde kapsadığını anlamak için üç temel ölçüt üzerinden değerlendirilir:

- Kesinlik (Precision) Model özetinde yer alan doğru kelimelerin oranını belirtmektedir
- Duyarlılık (Recall) Referans özetten doğru bir şekilde seçilen kelimelerin oranını ifade etmektedir:
- F1 Skoru Kesinlik ve duyarlılığın harmonik ortalamasını ölçmektedir.

ROUGE-2 (Recall-Oriented Understudy for Gisting Evaluation), metin özetleme sistemlerinin doğruluğunu ölçmek için kullanılan popüler bir metrik olup, referans özet ile model tarafından oluşturulan özet arasındaki iki kelimelik ardışık eşleşmeleri (bigram overlap) değerlendirir. Bu metrik, modelin yalnızca kelime seçiminde değil, aynı zamanda kelimelerin birbiriyle nasıl ilişkilendirildiği konusunda da doğruluğunu ölçer. ROUGE-2, metindeki bağlamsal tutarlılığı daha iyi değerlendirdiği için, ROUGE-1'e kıyasla daha kapsamlı bir ölçüm sağlar.

ROUGE-L (Recall-Oriented Understudy for Gisting Evaluation - Longest Common Subsequence), metin özetleme sistemlerinin doğruluğunu değerlendirmek için kullanılan bir metrik olup, referans özet ile model özet arasındaki en uzun ortak alt dizi (Longest Common Subsequence, LCS) üzerinden hesaplama yapmaktadır. ROUGE-L, kelimelerin sırasını ve bağlamını dikkate alarak, bir özetin referans metinle ne kadar örtüştüğünü ölçmektedir. Bu özellik, model özetinin hem içeriği hem de yapısal tutarlılığı açısından önemli bilgiler sağlamaktadır.

ROUGE-L, kelime sırasını ve bağlamı dikkate aldığı için, diğer ROUGE metriklerinden daha esnek bir şekilde performansı ölçmektedir. Kelimelerin doğru sırada yer alıp almadığını değerlendirmek, özetleme sistemlerinin doğal dil üretimindeki başarısını analiz etmek açısından kritik önem arz eder.

Avantajları:

- Kelimelerin sırasını dikkate alarak bağlamı değerlendirir.
- Tek kelimelik (unigram) ya da çift kelimelik (bigram) eşleşmelere kıyasla, daha esnek bir yapı sunar.

4.3.5 Çıktı Formatı

Sonuçlar, her iki özetleme modeli için ROUGE skorları değerlerini içerecek şekilde çıktı olarak raporlanacaktır. Kullanıcı bu sonuçları metin dosyası veya diğer formatlarda kaydedebilir.

4.4 Kısıtlamalar ve Sınırlamalar

4.4.1 Dil modeli kısıtlamaları

T5-Large ve SentenceTransformer modelleri, önceden belirlenmiş dil ve metin türlerine dayalı olarak eğitim almıştır. Türkçe metinlerin karmaşıklığı ve dil özellikleri, bazı dil modellerinin doğruluğunu sınırlayabilmektedir.

4.4.2 Donanım gereksinimleri

Yazılım, büyük dil modelleri ve çok sayıda metin verisi ile çalışırken, güçlü bir işlemci ve GPU gerektirir. Bu durum, daha düşük donanım kapasitesine sahip sistemlerde işlem sürelerinin uzamasına yol açabilmektedir.

4.5 Tasarım Süreci

Bu bölümde, Türkçe özetleme modeli geliştirme sürecindeki adımlar sistematik bir şekilde ele alınmıştır. Tasarım süreci, problemin analizi, sistem gereksinimlerinin belirlenmesi, model tasarımı ve test aşamalarını içermektedir.

4.5.1 Problemin analizi

Hedef: Türkçe metin özetleme görevinde mevcut modellerin performansını iyileştirmek.

Ana Sorunlar:

- Türkçe'nin eklemeli dil yapısının anlam kaybına yol açması.
- Özetlerin gramer ve anlam bütünlüğü açısından yetersiz bir seviyede olması şeklinde değerlendirilmesi.

Çözüm Yaklaşımı:

- Diktonomi yöntemi kullanılarak özetleme sürecinin bölümlere ayrılması.
- SentenceTransformer ile cümlelerin anlam taşıyan vektörlere dönüştürülerek model girdisi olarak optimize edilmesi.

4.5.2 Gereksinimlerin belirlenmesi

Fonksiyonel Gereksinimler:

- Metinleri anlam bütünlüğü korunarak özetleyebilme.
- Özetleme sırasında modelin Türkçe dil bilgisi kurallarına uygun çalışması.

Performans Ölçümü:

- ROUGE ve F1 skorları ile model performansının değerlendirilmesi.
- İnsan değerlendirmesi ile kalite kontrolü.

Sonuçların Görüntülenmesi:

- Özetlerin kullanıcı dostu bir formatta çıktı olarak sunulması.

4.3.3 Model tasarımı

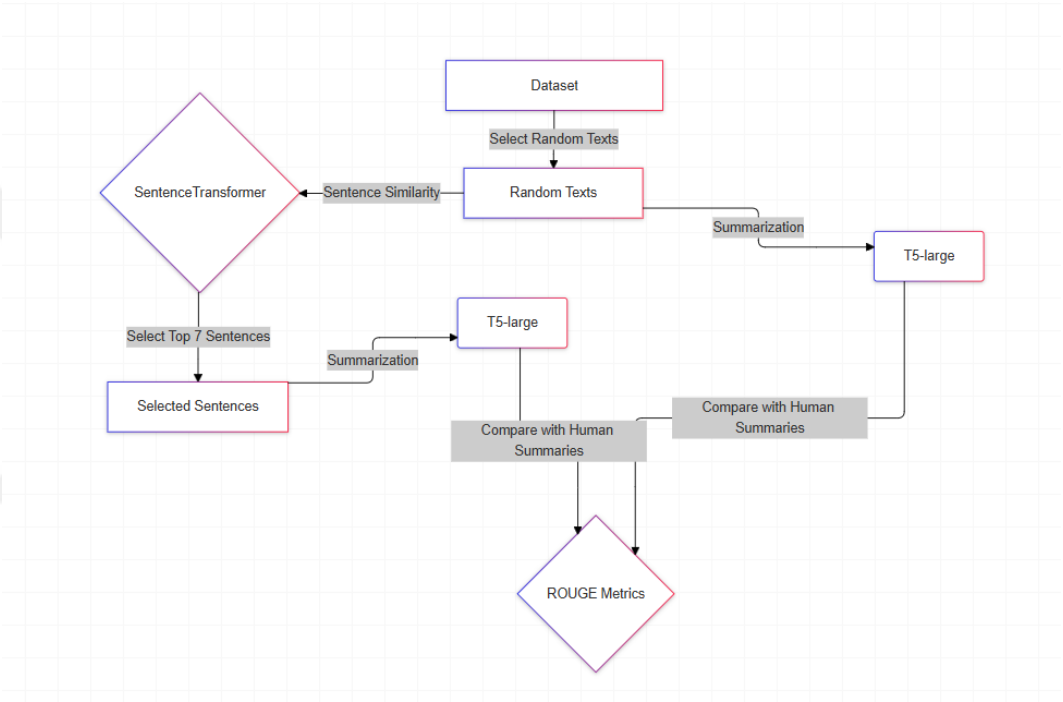
Diktonomi Yaklaşımı*:*

Önerilen model tasarımı, özetleme sürecini iki temel aşamaya ayıran diktonomi yaklaşımına dayanmaktadır:

1. **Cümle Önemi Analizi**: Bu aşamada, metindeki cümleler SentenceTransformer modeli olan "paraphrase-MiniLM-L6-v2" e (Reimers ve Gurevych, 2019) kullanılarak analiz edilmiş ve her bir cümlenin önemi sıralanmıştır. Bu sıralama, metin içerisindeki cümlelerin özetleme sürecindeki bağlamsal ve içeriksel değerine göre yapılmıştır.

2. **Bağlamlı Giriş Hazırlama**: Önemi yüksek olarak belirlenen yedi cümle seçilerek, T5-large modeline bağlamlı bir giriş olarak verilmiştir. Cümle sayısına kaynak verimliliği açısından karar verilmiştir. Bu süreç, modelin özetleme sürecinde daha anlamlı ve etkili sonuçlar üretmesine olanak tanımaktadır.

Tasarlanan model Şekil 4.1 aşağıdaki gibi gösterilmiştir.



Şekil 4.1: Diktonomi Yöntemiyle Tasarlanan Model İş Akışı

Dil Modeli*:*

Bu çalışmada, özetleme görevini gerçekleştirmek için önceden eğitilmiş T5-large (Raffel vd., 2020) modeli kullanılmıştır. Model, herhangi bir ince ayar yapılmaksızın, doğrudan ham hâliyle uygulanmıştır.

Girdi ve Çıktı Formatı*:*

- **Girdi:** Wikipedia alınan metinler ve değerlendirmede kullanılacak metin özetleri kullanılmıştır.
- **Çıktı:** Anlamlı, kısa ve Türkçe dil bilgisi kurallarına uygun bir özet ve performans skorları.

Bu tasarım, hem cümle düzeyinde analiz hem de dil modeli düzeyinde özetleme süreçlerini bir araya getirerek, özetlerin anlamlılık ve bağlamsal uygunluk açısından yüksek performans göstermesini amaçlamaktadır.

4.3.4 Mimari ve teknik yapı

Araçlar ve Teknolojiler:

- Python 3.9 programlama dili.
- Hugging Face kütüphaneleri: Transformers, Datasets
- SentenceTransformer (“paraphrase-MiniLM-L6-v2”) modeli.
- TensorFlow ve PyTorch Kütüphaneleri
- Rough Scorer ve Sklearn Kütüphaneleri

4.3.5 Geliştirme döngüsü

- **Prototip Geliştirme:** İlk model, küçük veri setleri üzerinde test edilmiştir.
- **İterasyonlar:** Model parametreleri, metriklere dayalı olarak optimize edilmiştir.
- **Entegrasyon:** Diktonomi yöntemi ile T5-large modeli bütünleşmiştir.
- **Performans Değerlendirme:** Ölçütlere uygunluk testleri gerçekleştirilmiştir.

Bu yazılımın tasarım süreci, metin özetleme işlemine başlamadan önce metnin cümlelerini değerlendirme ve modelin daha verimli çalışmasını sağlamak amacıyla problemi parçalara ayıran iki aşamalı bir yaklaşım benimsemektedir. Bu tasarımda, diktonomi (problemi parçalara ayırma) yaklaşımını kullanarak özetleme işlemi optimize edilmektedir.

4.4. Geliştirme Süreci

Bu bölüm, yazılım geliştirme sürecinde izlenen adımları ve kullanılan yaklaşımları ayrıntılandırmakta, karşılaşılan zorluklara ve çözümlere odaklanmaktadır.

4.4.2. Sistem Mimarisi

Sistem, "diktonomi" yaklaşımına uygun olarak iki ana bileşen üzerine inşa edilmiştir:

Ön Analiz Modülü: SentenceTransformer modeliyle anlam skorları oluşturulmuş ve metin önceliklendirme sağlanmıştır.

- Metin Özetleme Modülü: T5 modeli kullanılarak optimize özetleme algoritması geliştirilmiştir.

4.4.3. Geliştirme Adımları

4.4.3.1 Veri hazırlığı

- Metinler temizlenerek ön işleme adımları tamamlanmıştır.
- Önem skorlarını belirlemek için veri setleri etiketlenmiştir.

4.4.3.2 Model Kullanımı:

- SentenceTransformer ile metin anlam vektörleri oluşturulmuş ve bu skorlar üzerinde çalışılmıştır.
- T5 modeli, Türkçe metinlere uygun şekilde yeniden eğitilerek optimize edilmiştir.

4.4.3.3 Entegrasyon*:*

- SentenceTransformer ve T5 modelleri arasında sorunsuz veri aktarımı sağlanmıştır.
- Öncelikli cümleler, nihai özetleme için girdi olarak sunulmuştur.

4.4.3.4 Performans analizi

- Modelin başarısı ROUGE skorları kullanılarak değerlendirilmiştir.
- Elde edilen sonuçlar insan değerlendirmesiyle desteklenmiştir.

4.4.4. Karşılaşılan Zorluklar ve Çözümler

4.4.4.4 Hafıza ve hesaplama sınırları

Sorun:

T5-large modelinin büyük boyutu, bellek tüketimini ve hesaplama süresini önemli ölçüde artırmıştır. Model, özellikle büyük hacimli metinler işlendiğinde yüksek miktarda işlemci ve bellek kaynağı gerektirmiş, bu da hem modelin çalıştırılması sırasında zaman kaybına yol açmış hem de mevcut donanım kapasitesinin sınırlayıcı bir faktör haline gelmesine neden olmuştur. Bellek sınırlarının aşılması, modelin işlem yükünü artırarak eğitim ve değerlendirme süreçlerinin yavaşlamasına yol açmıştır.

Çözüm:

Bu sınırlamaları aşmak ve modelin performansını artırmak için birkaç strateji uygulanmıştır:

- **Veri Bölme Stratejisi:** Modelin bellek tüketimini ve işlem süresini azaltmak amacıyla, işlenecek veri seti daha küçük parçalara bölünerek farklı zaman dilimlerinde işlenmiştir. Bu yaklaşım, tüm veri setinin tek seferde modele verilmesinden kaynaklanan bellek sorunlarını ortadan kaldırmış ve donanımın mevcut kapasitesine uygun bir şekilde çalışmasına olanak sağlamıştır. Her bir veri parçası, bağımsız olarak modele sunulmuş ve işlem tamamlandıktan sonra sonuçlar birleştirilmiştir. Bu yöntem, modelin kaynak tüketimini önemli ölçüde optimize etmiş ve sürecin sürdürülebilirliğini artırmıştır.

- **TPU Kullanımı:** Modelin hesaplama yükünü optimize etmek ve işlem sürelerini kısaltmak amacıyla Google Colab ortamında Tensor Processing Unit (TPU) teknolojisi kullanılmıştır. TPU, yüksek performanslı paralel hesaplama yetenekleri sayesinde modelin büyük boyutlu parametrelerini hızlı bir şekilde işlemiş ve GPU'ya kıyasla daha verimli bir çözüm sunmuştur. TPU'nun bu avantajı, modelin eğitim ve özetleme görevlerinde işlem sürelerini önemli ölçüde kısaltmıştır.

Bu stratejiler bir araya getirildiğinde, T5-large modelinin büyük boyutundan kaynaklanan bellek ve hesaplama sorunları minimize edilerek, hem donanım kaynaklarının etkin kullanımı sağlanmış hem de modelin daha verimli çalışması mümkün olmuştur. Özellikle veri setinin parçalara ayrılarak işlenmesi, büyük ölçekli modellerin kısıtlı donanım ortamlarında etkili bir şekilde kullanılabileceğini göstermektedir.

4.4.4.5 Performans değerlendirmesi

Sorun:

Modelin ürettiği özetlerin doğruluğunu ve anlamlılığını değerlendirme süreci, insan gözlemine dayalı bir yaklaşımla yürütüldüğünde çeşitli zorluklar içermektedir. İnsan gözlemi, öznel bir süreç olması nedeniyle değerlendiriciler arasında tutarlılık eksikliği yaşanabilmekte ve bu durum, sonuçların güvenilirliğini etkileyebilmektedir. Ayrıca, büyük ölçekli veri setlerinde insan gözlemine dayalı değerlendirme, zaman alıcı ve yorucu bir süreç hâline gelmiştir. Bu sorunlar, özetlerin niceliksel bir değerlendirme ile desteklenmesini gerektirmiştir.

Çözüm:

İnsan gözlemine dayalı değerlendirme sürecinin daha sistematik ve güvenilir bir hâle getirilmesi amacıyla ROUGE metrikleri kullanılarak destekleyici bir analiz yöntemi benimsenmiştir. Bu kapsamda şu adımlar izlenmiştir:

ROUGE Metrikleri ile Niceliksel Değerlendirme: İnsan gözlemlerinin yanına, referans özetler ile model tarafından üretilen özetler arasındaki örtüşmeyi ölçmek için ROUGE-1, ROUGE-2 ve ROUGE-L gibi metrikler uygulanmıştır. Bu metrikler, modelin doğru kelimeleri, kelime çiftlerini ve uzun dizilimleri ne kadar iyi tahmin ettiğini nicel olarak değerlendirme imkânı sunmuştur.

Hibrit Değerlendirme Yaklaşımı: ROUGE metriklerinden elde edilen niceliksel sonuçlar, insan gözlemlerini desteklemek amacıyla kullanılmıştır. Bu hibrit yaklaşım, değerlendirme sürecine nesnellik kazandırmış ve insan gözlemine dayalı kararların doğruluğunu artırmıştır. Örneğin, insan gözlemine göre anlamlı bulunan bir özetin ROUGE skorlarının da yüksek olması, sonuçların tutarlılığını pekiştirmiştir.

Değerlendiriciler Arası Tutarlılık: İnsan gözlemine dayalı değerlendirmelerde değerlendiriciler arasında tutarlılık sağlanması için ROUGE metriklerinden elde edilen

sonuçlar bir referans noktası olarak sunulmuştur. Bu, farklı değerlendiricilerin aynı standartlara göre karar vermesine yardımcı olmuştur.

Zaman ve Kaynak Verimliliği: ROUGE metriklerinin kullanımı, insan gözlemine dayalı değerlendirme sürecini hızlandırmış ve büyük ölçekli veri setlerinde daha kısa sürede daha tutarlı sonuçlar elde edilmesini sağlamıştır.

Bu yaklaşım, insan gözleminin sağladığı derinlemesine değerlendirme ile ROUGE metriklerinin sunduğu nesnelliği birleştirerek daha kapsamlı ve güvenilir bir analiz yöntemi geliştirmiştir. Sonuç olarak, insan gözlemine dayalı değerlendirmenin zorlukları büyük ölçüde aşılmış ve özetleme sisteminin performansı hem niteliksel hem de niceliksel olarak daha etkili bir şekilde değerlendirilmiştir

4.5. Test ve Doğrulama

Bu çalışmanın test ve doğrulama aşamasında, Türkçe metin özetleme için kullanılan T5 modelinin performansı, iki farklı özetleme yöntemiyle değerlendirilmiştir. Test sürecinde, T5 modeline verilen ham metin ve SentenceTransformer ile işlenmiş metinler üzerinden üretilen özetlerin kalitesi, ROUGE-1, ROUGE-2 ve ROUGE-L gibi metriklerle karşılaştırılmıştır. Bu aşamada elde edilen özetler ve karşılaştırmalı ROUGE skorları, modelin özetleme başarısını bağımsız veri kümeleri üzerinde doğrulamayı sağlamış ve sonuçların istatistiksel güvenilirliğini artırmıştır.

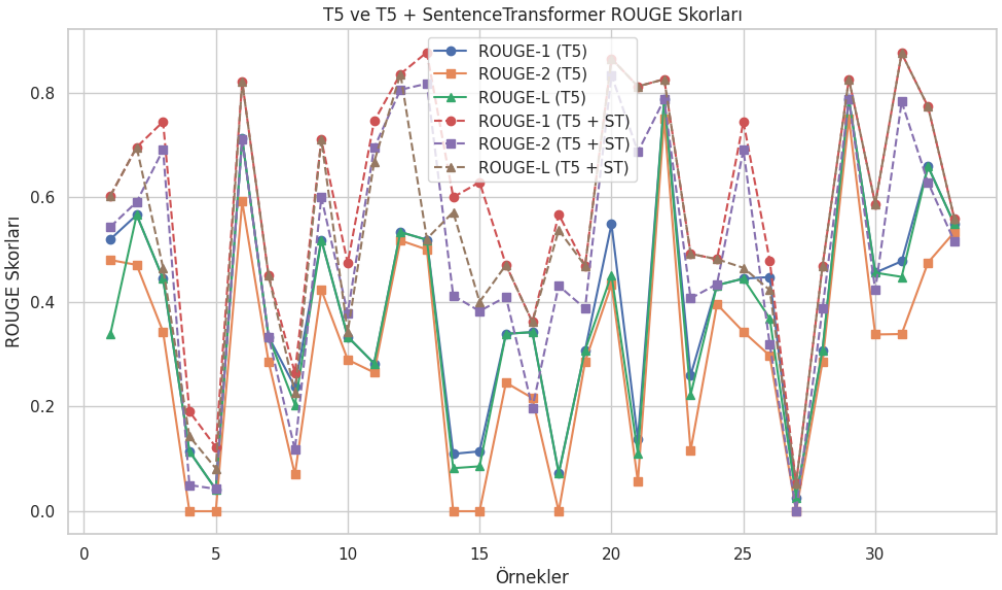
4.6 Sonuçlar ve Değerlendirme

4.6.1 Başarı değerlendirmesi

Bu çalışmada, Türkçe metin özetleme performansı, T5 modelinin özetlerinden elde edilen ROUGE skorlarıyla değerlendirilmiştir. Doğrulama setindeki 262 örnek üzerinde yapılan analizde, özellikle bazı örneklerde SentenceTransformer (ST) ile işlem gören özetlerin, yalnızca T5 modelinin özetlerine kıyasla belirgin şekilde daha yüksek ROUGE skorları elde ettiği gözlemlenmiştir. Özellikle, ROUGE-1, ROUGE-2 ve ROUGE-L metriklerinde SentenceTransformer ile iyileştirilmiş özetler daha iyi sonuçlar vermiştir.

Tablo 1 de gözlemleneceği üzere, bazı örneklerde ROUGE-1 skoru T5 modelinden 0.52'den 0.60'a, ROUGE-2 skoru ise 0.48'den 0.54'e çıkmıştır. Bu artış, SentenceTransformer'ın özetleme kalitesine katkısını göstermektedir. Bununla birlikte, bazı örneklerde SentenceTransformer'ın ROUGE değerlerinin beklentilerin çok üzerinde olduğu gözlemlenmiştir. Özellikle, bazı özetlerin ROUGE-L skoru 0.82 seviyelerine ulaşmış ve bu değer T5 modelinin sunduğu sonuçlardan belirgin şekilde daha yüksek olmuştur.

Bu durumu açıklamak için, SentenceTransformer'ın metinler arasındaki anlam benzerliğini daha iyi yakalayabilmesi ve T5 modelinin üretmeye çalıştığı özetlerin içeriğini daha tutarlı bir şekilde yansıtabilmesi etkili olabilir. Ancak, bazı durumlarda yüksek ROUGE skorları, modelin sadece benzer kelimeleri veya yüzeysel benzerlikleri yakaladığını, anlamlı içerik sağlama açısından sınırlı kalabileceğini de göstermektedir. Sonuç olarak, SentenceTransformer ile geliştirilen özetlerin daha yüksek ROUGE skorları üretmesi, modelin daha anlamlı özetler ürettiğini gösteriyor olabilir, ancak bu durum dikkatlice değerlendirilmelidir.



Şekil 4.2: MT5-Large ve SentenceTransformer+T5-Large Modelleri Performans Sonuçları

Şekil 4.2 incelendiğinde, SentenceTransformer'ın (ST) kullanımı, özetleme performansını bazı durumlarda artırdığı görülmektedir. ROUGE-1, ROUGE-2 ve ROUGE-L metriklerinde, SentenceTransformer'ın etkisi bazı örneklerde olumlu olmuştur. Örneğin, ROUGE-1 (T5) skoru 0.52'den 0.60'a yükselmiş, aynı şekilde ROUGE-2 (T5) skoru da 0.48'den 0.54'e çıkmıştır. ROUGE-L skorları da benzer şekilde iyileşmiştir.

Özellikle, bazı örneklerde ROUGE-1, ROUGE-2 ve ROUGE-L değerlerinde 0.712, 0.591, 0.712 gibi yüksek sonuçlar elde edilmiştir. Bu sonuçlar, SentenceTransformer'ın bazı metin içerisindeki önemli cümleleri daha iyi tespit edebildiğini ve bu sayede T5 modelinin özetleme başarısını artırdığını göstermektedir.

Şekil 4.2 de görülen bazı örneklerde ise SentenceTransformer'ın kullanılması ile ROUGE skorları oldukça düşük kalmıştır. Örneğin, ROUGE-1 (T5) skoru 0.04 gibi çok düşük bir değere sahip örnekler bulunmaktadır. Bu tür düşük skorlar, özetlemenin kalitesiz veya yetersiz olduğu, ya da metnin karmaşıklığından kaynaklanan zorluklar

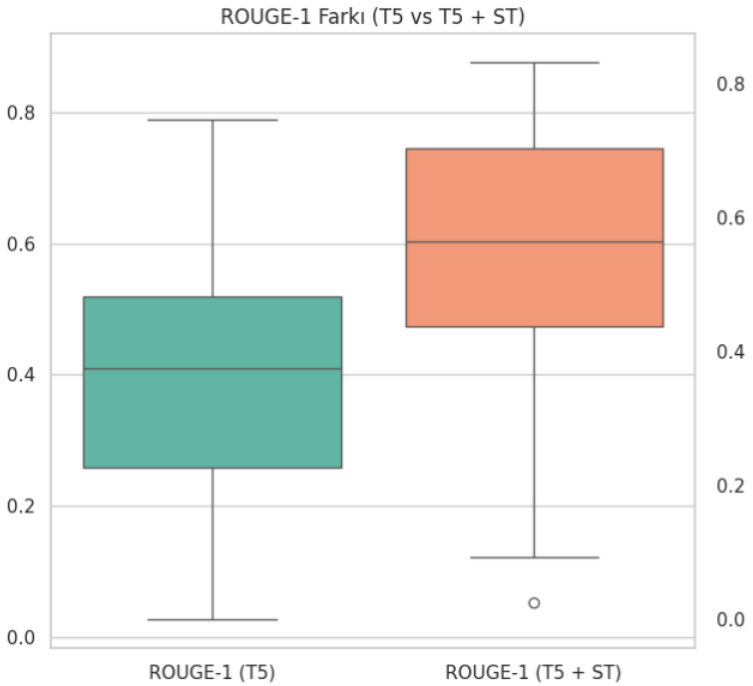
olduğunu gösterebilir. ROUGE-2 ve ROUGE-L metriklerinde de benzer şekilde düşük değerler gözlemlenmiştir.

Bazı örneklerde, SentenceTransformer'ın etkisi sınırlı olmuş ve ROUGE-1 skoru, sadece 0.05'e yükselmiştir. Bu, SentenceTransformer'ın yalnızca sınırlı ölçüde fayda sağladığı ya da seçilen cümlelerin modelin özetleme performansını önemli ölçüde iyileştiremediği durumları işaret edebilir.

Bazı örneklerde, SentenceTransformer ile yapılan işlem sonucunda özetlemenin kalitesinde belirgin bir iyileşme gözlemlenmiştir. Örneğin, ROUGE-1 (T5) skoru 0.44'ten 0.74'e, ROUGE-2 (T5) skoru 0.34'ten 0.69'a, ROUGE-L (T5) skoru ise 0.44'ten 0.47'ye çıkmıştır. Bu tür artışlar, SentenceTransformer'ın belirli metinlerde anlamlı cümleleri seçerek, özetin kalitesini artırmaya yardımcı olduğunu ve modelin daha doğru özetler üretmesini sağladığını göstermektedir.

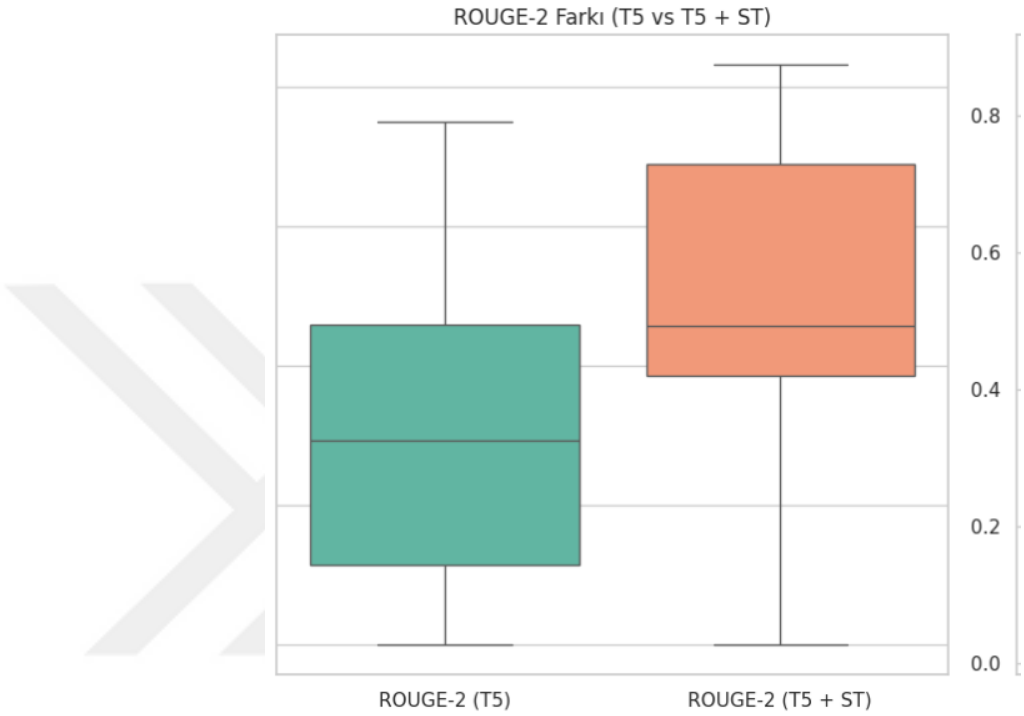
Bazı örneklerde ROUGE skorlarında dalgalanma görülmektedir. Örneğin, ROUGE-1 (T5) skoru 0.71 gibi yüksek bir değerden 0.11'e kadar düşerken, ROUGE-2 ve ROUGE-L metriklerinde de benzer bir düşüş yaşanmıştır. Bu tür dalgalanmalar, bazı metinlerin özetlenmesinin özellikle zor olduğuna işaret edebilir. Bu tür metinler genellikle karmaşık yapılar, T5 modelinin kelime temsilinin SentenceTransformer kelime temsili ile uyumsuzluğunu ve çok sayıda teknik terim veya anlamı zor cümleler içerebilir, bu da modelin başarısını etkileyebilmektedir.

Şekil 4.2 de özellikle dikkat çeken yüksek performanslı örnekler, ROUGE-1 ve ROUGE-2 metriklerinde sırasıyla 0.79 ve 0.75 gibi yüksek değerler elde edilmiştir. Bu örnekler, SentenceTransformer ve T5 modelinin birleşimiyle elde edilen yüksek kaliteli özetleri temsil etmektedir. Bu, SentenceTransformer'ın anlamlı ve önemli cümleleri doğru bir şekilde özetlere dâhil edebilmesinin bir göstergesidir.



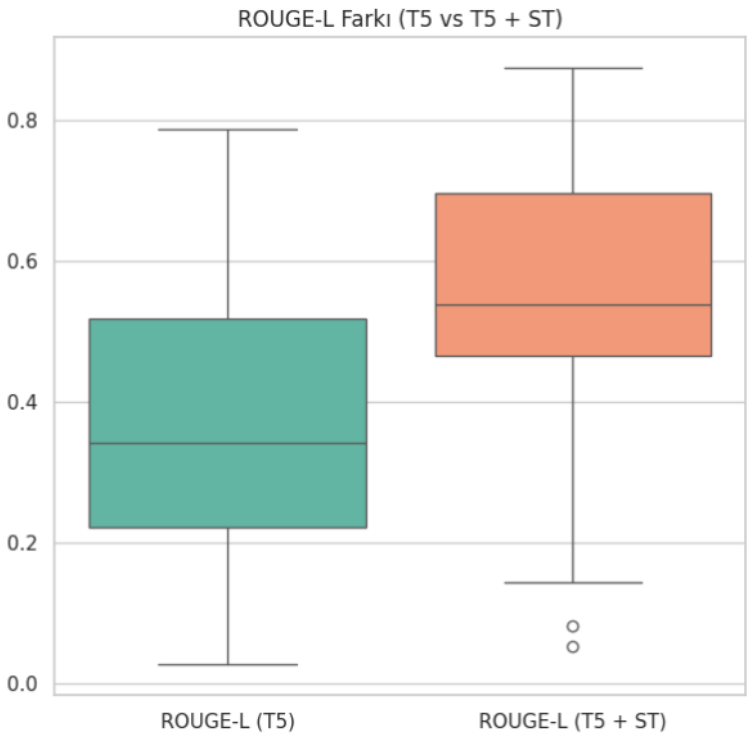
Şekil 4.3: İyi Performans Gösteren Örneklerin ROUGE-1 (T5) ile ROUGE-1 (T5 + ST) Karşılaştırması

Şekil 4.3' den anlaşılacağı gibi kosinüs benzerliklerinin kullanımda iyi performans gösteren örneklerin T5-Large'a göre ortalamada belirgin iyileşme gözlemleniyor. Bu, SentenceTransformer'ın metnin genel anlamını ve önemli cümlelerini daha iyi yakalayarak özetleme kalitesini artırdığını göstermektedir. SentenceTransformer, özellikle kelime ve cümle düzeyindeki ilişkileri daha etkili bir şekilde modelleyerek, özetin anlam bütünlüğünü iyileştirmekte ve dolayısıyla ROUGE-1 skorlarında önemli bir artışa yol açmaktadır.



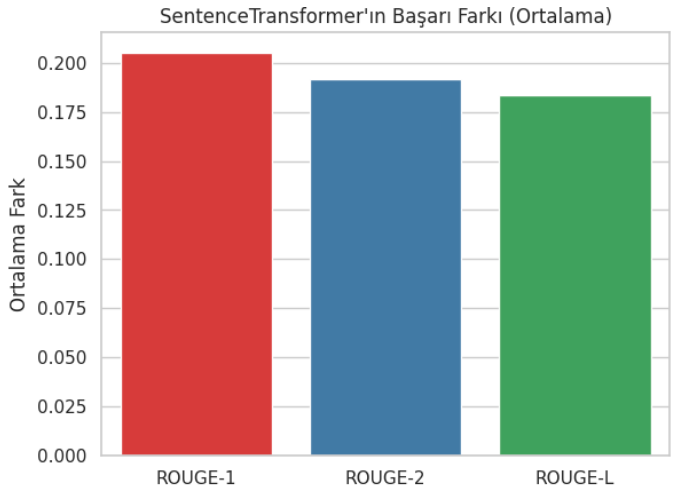
Şekil 4.4: İyi Performans Gösteren Örneklerin ROUGE-2 (T5) ile ROUGE-2 (T5 + ST) Karşılaştırması

ROUGE-2 skorlarında ise (Şekil 4.4), kosinüs benzerliklerinin kullanımda iyi performans gösteren örneklere etkisi daha ılımlı olmuştur. Bu skor, genellikle iki kelime arasındaki n-gram benzerliğini ölçtüğü için, anlam düzeyindeki iyileşmeler kadar belirgin değildir. Ancak, yine de SentenceTransformer'ın etkisi görülmektedir, çünkü bazı örneklerde büyük artışlar kaydedilmiştir. Bu durum, SentenceTransformer'ın, özellikle daha uzun ve anlamlı n-gram'lar için daha fazla bilgi sunarak, modelin kelime ve ifade ilişkilerini daha etkili bir şekilde öğrenmesine olanak tanıdığına işaret etmektedir.



Şekil 4.5: ROUGE-L (T5) ile ROUGE-L (T5 + ST) Karşılaştırması

Şekil 4.5'den de gözlemleneceği gibi, ROUGE-L (Longest Common Subsequence) skoru, cümleler arasındaki en uzun ortak alt diziyi ölçer ve metnin genel anlamının yanı sıra yapısal bütünlüğünü de dikkate alır. Burada gözlemlenen orta düzeydeki iyileşmeler**,** SentenceTransformer'ın metnin uzun vadeli yapısal ve anlamsal ilişkilerini anlamada daha az etkili olduğunu düşündürmektedir. Bununla birlikte, SentenceTransformer hala özetin bağlamını geliştirmekte ve modelin sonuçlarını iyileştirmektedir, ancak bu etki ROUGE-1 ve ROUGE-2'ye kıyasla daha sınırlıdır**.**



Şekil 4.6: İyi Performans Gösteren Örneklerin SentenceTransformer'ın Başarı Farkı (Ortalama)

Genel olarak (Şekil 4.6), SentenceTransformer'ın T5 modelinin özetleme performansını iyileştirmede olumlu bir etkisi olduğu söylenebilir. Özellikle ROUGE-1'deki belirgin artış, SentenceTransformer'ın özetleme sürecine katkılarının anlamın doğruluğunu artırmaya yönelik olduğunu gösteriyor. ROUGE-2'deki daha küçük iyileşmeler, bu katkının daha çok kelime düzeyindeki n-gram benzerliklerine dayalı olduğunu, ROUGE-L'deki daha düşük iyileşmeler ise metnin yapısal bütünlüğü üzerinde yapılan iyileştirmelerin sınırlı olduğunu göstermektedir.

Bu bulgular, SentenceTransformer'ın özellikle anlam düzeyindeki iyileştirmeler konusunda güçlü olduğunu ve metnin özünü daha iyi kavrayarak özetlemeyi daha etkili hale getirdiğini, ancak yapısal ve n-gram bazlı iyileştirmelerde etkisinin daha sınırlı kaldığını ortaya koymaktadır.

4.6.1 T-Testi

Bu bölümde SentenceTransformer modelinin özetleme performansı üzerindeki etkisini bilimsel bir bakış açısıyla değerlendirebilmek için istatistiksel bir analiz yapılmıştır. Bu doğrultuda, t-testi istatistiksel test olarak kullanılmış,

SentenceTransformer'ın metin özetleme sürecine dâhil edilmesinin, modelin performansı üzerinde anlamlı bir fark yaratıp yaratmadığı araştırılmıştır.

Bu çalışmada, SentenceTransformer modelinin özetleme başarısı üzerindeki etkisini belirlemek amacıyla, iki bağımlı örnek grup arasında karşılaştırma yapılacaktır: bir grupta SentenceTransformer kullanılacak, diğer grupta ise kullanılmayacaktır. Bu karşılaştırma, özetleme sonuçları arasında anlamlı bir fark olup olmadığını tespit etmeye yönelik olacaktır.

Bu istatistiksel testin amacı, aşağıdaki hipotezleri test etmektir:

- **Null Hipotezi (H_0):** SentenceTransformer kullanımı ile elde edilen özetler ile kullanılmadığı özetler arasındaki performans farkı istatistiksel olarak anlamlı değildir.

- **Alternatif Hipotez (H_1):** SentenceTransformer kullanımı ile elde edilen özetler ile kullanılmadığı özetler arasındaki performans farkı istatistiksel olarak anlamlıdır.

Bu hipotez testi, SentenceTransformer modelinin özetleme başarısını iyileştirip iyileştirmediğini, yani modelin etkisinin istatistiksel olarak anlamlı olup olmadığını belirlemek için kullanılacaktır. Test sonucunda elde edilecek p-değeri, hipotezler arasındaki farkın istatistiksel anlamlılık düzeyini gösterecek ve bu bulgular, araştırmanın sonucuna dayalı olarak doğru çıkarımlar yapılmasına olanak sağlayacaktır.

4.6.1 T-Testi Sonuçları

4.6.1.1 ROUGE-1 sonucu

Test sonucunda değerler t-istatistiği: 8.97, p-değeri: $6.03 \times 10{-}17$ olarak elde edilmiştir. Bu sonuç, T5 ve T5 + SentenceTransformer özetleme yöntemleri arasındaki farkın istatistiksel olarak anlamlı olduğunu göstermektedir. P-değeri son derece küçük (0.05'ten çok daha küçük) olduğu için, Null Hipotez (yani, iki yöntem arasındaki farkın

rastlantısal olduğu) reddedilir. Sonuçlar, SentenceTransformer kullanımının ROUGE-1 skorunu belirgin şekilde iyileştirdiğini ve bu iyileştirmenin rastlantısal olmadığını ortaya koymaktadır. Bu durum, SentenceTransformer'ın, metnin anlamını daha doğru bir şekilde yansıtan önemli cümleleri seçerek özetleme performansını artırdığını gösterir.

4.6.1.2 ROUGE-2 sonucu

Test sonucunda değerler t-istatistiği: 9.57, p-değeri: 8.31×10^{-19} olarak elde edilmiştir. Benzer şekilde, ROUGE-2 metrik ölçümünde de p-değeri oldukça düşük olup, farkın istatistiksel olarak anlamlı olduğunu doğrulamaktadır. SentenceTransformer'ın T5'in özetleme performansını daha da iyileştirdiği ve bu iyileştirmenin rastlantısal olmadığını göstermektedir. ROUGE-2 metriği, özetlerin n-gram benzerliğini ölçtüğü için, SentenceTransformer'ın metnin anlamını seçme başarısının n-gram düzeyinde de özetin kalitesine önemli bir katkı sağladığını ifade eder.

4.6.1.3 ROUGE-L sonucu

Test sonucunda değerler t-istatistiği: 10.80, p-değeri: 9.94×10^{-23} olarak elde edilmiştir. ROUGE-L metriği, özetin anlam bütünlüğünü ve cümle yapısının ne kadar benzer olduğunu ölçerken, SentenceTransformer'ın özetleme performansını daha da iyileştirdiğini gösteren en güçlü sonuçlardan birini ortaya koymaktadır. p-değeri son derece düşük, bu da farkın anlamlı olduğunu ve iyileştirmenin rastlantısal olmadığını açıkça göstermektedir. SentenceTransformer, özellikle metnin bağlamını daha iyi yakalayarak, özetlerin daha doğal ve anlamlı olmasına yardımcı olmuştur. Bu, metnin anlamını koruma ve doğru bağlamda anlamlı özetler üretme açısından önemli bir başarıdır.

4.6.2 T-Testi Sonuçları Genel Değerlendirme

Tüm üç ROUGE metriği (ROUGE-1, ROUGE-2 ve ROUGE-L) üzerinde yapılan testler, T5 + SentenceTransformer kombinasyonunun T5'e kıyasla anlamlı şekilde bazı örneklerde iyi performans gösterdiğini ortaya koymaktadır. Bu sonuçlar, özetlenecek

metnin cümleleri arasında kosinüs benzerliği en yüksek seviyede olan cümlelerin metnin anlamını doğru bir şekilde çıkararak, özetlerin kalitesini artıran önemli bir katkı sağladığını açıkça gösteriyor. Özellikle bağlamı ve n-gram yapısını koruyarak yapılan iyileştirmeler, özetleme süreçlerinin daha doğru ve anlamlı hale gelmesine olanak tanımaktadır.

4.6.3 Sonuç

Bu bulgular, kosinüs benzerlik tabanlı içerik seçimi ile özetleme görevlerinde T5 modeline bütünleştirilerek bazı örneklere performans artışı sağladığını ve bu artışın istatistiksel olarak anlamlı olduğunu göstermektedir. Bu iyileştirmeler, Türkçe özetleme gibi dil işleme görevlerinde daha doğru ve anlamlı metinler üretmek için güçlü bir adım atıldığını ortaya koymaktadır.

Özellikle bazı örneklerde, SentenceTransformer'ın kullanımı ile özetlerin ROUGE skorları önemli ölçüde artmıştır. ROUGE-1 ve ROUGE-2 skorlarında sırasıyla 0.52'den 0.60'a, 0.48'den 0.54'e kadar iyileşmeler gözlemlenmiştir. Daha yüksek performanslı örneklerde, ROUGE-1 skoru 0.71'e, ROUGE-2 skoru ise 0.75'e kadar yükselmiştir. Bu iyileşmeler, SentenceTransformer'ın metin içindeki anlamlı ve önemli cümleleri daha iyi tespit ederek, T5 modelinin özetleme başarısını artırdığına işaret etmektedir.

Ancak, tüm örneklerde benzer iyileşmeler gözlemlenmemiştir. Bazı metinlerde, kosinüs benzerlik tabanlı içerik seçimi ile özetleme etkisi sınırlı kalmış ve ROUGE skorlarında düşüşler yaşanmıştır. Örneğin, bazı örneklerde ROUGE-1 skoru sadece 0.04'e kadar düşerken, ROUGE-2 ve ROUGE-L metriklerinde de düşük sonuçlar alınmıştır. Bu, özetlemenin kalitesiz veya yetersiz olduğu, ya da metnin karmaşıklığından kaynaklanan zorlukların etkisiyle olduğu varsayılabilir. Ayrıca, metinlerdeki teknik terimler veya anlam karmaşıklıkları, modelin özetleme performansını olumsuz etkileyebilmiştir.

Kosinüs benzerlik tabanlı içerik seçimi ile özetleme yüksek performans gösterdiği örneklerde, anlam düzeyinde önemli iyileşmeler sağlanırken, yapısal ve n-gram bazlı

iyileştirmelerde daha sınırlı bir etki gözlemlenmiştir. Özellikle ROUGE-1 ve ROUGE-2 metriklerinde yapılan iyileşmeler, SentenceTransformer'ın kelime ve cümle düzeyindeki ilişkileri daha iyi modelleyebilmesinden kaynaklanmaktadır. ROUGE-L skorlarında ise daha düşük iyileşmeler elde edilmiştir, bu da metnin yapısal bütünlüğü üzerinde yapılan iyileştirmelerin sınırlı kaldığını göstermektedir.

Sonuç olarak, bu çalışma, SentenceTransformer modelinin, özellikle anlam düzeyindeki iyileştirmeler konusunda güçlü olduğunu ve metnin özünü daha iyi kavrayarak özetlemeyi daha etkili hale getirdiğini ortaya koymaktadır. Ancak, yapısal bütünlük ve n-gram bazlı benzerliklerdeki sınırlı iyileşmeler, bu yaklaşımın her metin türü için aynı derecede etkili olamayabileceğini göstermektedir. Gelecekte, SentenceTransformer'ın yapısal iyileştirmelere daha fazla katkı sağlaması amacıyla, modelin daha kapsamlı bir şekilde eğitilmesi veya yapısal unsurların dikkate alındığı ek yöntemlerin geliştirilmesi faydalı olabilir.

4.7 Gelecek Çalışmalar ve Öneriler

Gelecekteki çalışmalar, Kosinüs Benzerlik Tabanlı İçerik Seçimi modelinin performansını daha geniş veri setleri üzerinde test ederek modelin genelleme yeteneğini daha iyi değerlendirebilir. Bu tür testler, modelin yalnızca belirli bir veri seti veya konu üzerinde değil, farklı türdeki metinler ve çeşitli konularda da nasıl performans gösterdiğini gözler önüne serecektir. Bu, modelin daha geniş bir uygulama yelpazesinde geçerliliğini anlamamıza yardımcı olacaktır.

Farklı dillerde yapılan analizler de önemli bir katkı sağlayabilir. Şu anki çalışma Türkçe metinler üzerinde odaklanmış olsa da, modelin farklı dillerde nasıl çalıştığını görmek, modelin dil bağımsız performansını değerlendirme açısından kritik öneme sahiptir. Örneğin, İngilizce, Fransızca veya Almanca gibi dillerde de benzer analizler yaparak, modelin dil farklılıklarına ne kadar duyarlı olduğunu ve bu dillerde de ne kadar etkili olduğunu incelemek, modelin evrensel başarısını değerlendirmek için önemli bir

adım olacaktır. Böylece, modelin yalnızca Türkçe değil, çok dilli özetleme görevlerinde de başarı sağlaması için gerekli iyileştirmeler yapılabilir.

Modelin farklı özetleme görevlerine uygulanması da gelecekteki araştırmalara yön verebilir. Örneğin, haber metinleri, bilimsel metinler ve sosyal medya içerikleri gibi farklı metin türleri, özetleme görevlerinin farklı gereksinimlerine sahip olabilir. Haber metinleri genellikle güncel, kısa ve öz bilgilere dayanırken, bilimsel metinler daha karmaşık, uzun ve teknik terimler içerir. Sosyal medya içerikleri ise dilsel olarak daha serbest, dil bilgisel olarak daha esnek ve bazen anlam karmaşıklığı barındıran metinlerdir. Her bir metin türü, özetleme sürecinde farklı zorluklar ve fırsatlar yaratır. Kosinüs Benzerlik Tabanlı İçerik Seçiminin bu tür metinler üzerindeki performansı, modelin çok çeşitli dilsel bağlamlarda nasıl çalıştığı konusunda önemli bilgiler verecektir. Farklı metin türlerine yönelik testler, modelin bu metin türleriyle ne kadar etkili başa çıkabildiğini ve bu türlerin özetlenmesinde ne kadar başarılı olduğunu anlamamıza yardımcı olacaktır.

Bununla birlikte, bu tür farklı metin türlerine ve dillere yönelik testler, modelin kapsamını genişleterek daha evrensel bir başarıya ulaşmasını sağlayabilir. Gelecekteki çalışmalar, modelin evrensel başarısını sadece dil ve konu çeşitliliği bağlamında değil, aynı zamanda farklı özetleme görevlerinde sağladığı başarıyı da gözlemleyerek modelin genel başarısını değerlendirmemize yardımcı olacaktır. Bu, hem modelin genel kullanılabilirliğini artırabilir hem de özetleme teknolojisinin daha geniş bir kitleye ve farklı alanlara hitap etmesini sağlayabilir.

KAYNAKÇA

Abdel-Basset, M., Moustafa, N., Hawash, H., Ding, W., Abdel-Basset, M., Moustafa, N. & Ding, W. (2022). Supervised Deep Learning for Secure Internet of Things. *Deep Learning Techniques for IoT Security and Privacy*, 131-166.Springer. https://link.springer.com/chapter/10.1007/978-3-030-89025-4_5

Arisoy, E., Saraçlar, M., Roark, B., & Shafran, I. (2011). Discriminative language modeling with linguistic and statistically derived features. *IEEE Transactions on Audio, Speech, and Language Processing*, *20*(2), 540–550. https://doi.org/10.1109/TASL.2011.2162323

Arqawi, S. M., Zitawi, E. A., Rabaya, A. H., Abunasser, B. S., & Abu-Naser, S. S. (2022). Predicting university student retention using artificial intelligence. International Journal of Advanced Computer Science and Applications, *13*(9), 1–10. https://doi.org/10.14569/IJACSA.2022.0130937

Avrupa Komisyonu. (2019). *Avrupa Yeşil Mutabakatı*. Avrupa Birliği Yayın Ofisi. https://eur-lex.europa.eu/legal-content/TR/TXT/?uri=CELEX%3A52019DC0640

Azlah, M. A. F., Chua, L. S., Rahmad, F. R., Abdullah, F. I., & Wan Alwi, S. R. (2019). Review on techniques for plant leaf classification and recognition. *Computers*, *8*(4), 77. https://doi.org/10.3390/computers8040077

Bakan, C. T., & Yakut, S. (2024). Graf teorisi ve Malatya merkezilik algoritmasına dayalı Haber metinlerinin özetlemesi. *Bilişim Teknolojileri Dergisi*, *17*(3), 189–198. https://doi.org/10.17671/gazibtd.1463107

Belhaouari, S. B., & Islam, A. (2021). Deep learning in healthcare. *In Multiple perspectives on artificial intelligence in healthcare* (pp. 155–168). Springer.

Bengio, Y., Lecun, Y., & Hinton, G. (2021). Deep learning for AI. *Communications of the ACM*, *64*(7), 58–65. https://doi.org/10.1145/3448250

Bengio, S., Vinyals, O., Jaitly, N., & Shazeer, N. (2015). Scheduled sampling for sequence prediction with recurrent neural networks. *Advances in Neural Information Processing Systems*, *28*, 1171–1179. https://proceedings.neurips.cc/paper/2015/hash/e995f98d56967d946471af29d7bf99f1-Abstract.html

Berral-Garcia, J. L. (2018). When and how to apply statistics, machine learning and deep learning techniques. *In 2018 20th International Conference on Transparent Optical Networks (ICTON)* (pp. 1–4). IEEE. https://doi.org/10.1109/ICTON.2018.8473910

Dai, A. M., & Le, Q. V. (2015). Semi-supervised sequence learning. *Advances in Neural Information Processing Systems*, *28*, 3079–3087.

https://proceedings.neurips.cc/paper/2015/hash/7137debd45ae4d0ab9aa953017286b20-Abstract.html

Debelee, T. G., Kebede, S. R., Schwenker, F., & Shewarega, Z. M. (2020). Deep learning in selected cancers' image analysis—a survey. *Journal of Imaging*, *6*(11), 121. https://doi.org/10.3390/jimaging6110121

Ding, S., Zhang, N., Xiaoli, X., Guo, L., & Zhang, J. (2015). Deep extreme learning machine and its application in EEG classification. *Mathematical Problems in Engineering*, *2015*, 1–11. https://doi.org/10.1155/2015/129021

Engelen, J. E. v., & Hoos, H. H. (2019). A survey on semi-supervised learning. *Machine Learning*, *109*(2), 373–440. https://link.springer.com/article/10.1007/s10994-019-05855-6

Faradonbe, S. M., Safi-Esfahani, F., & Karimian-kelishadrokhi, M. (2020). A review on neural Turing machine (NTM). *SN Computer Science*, *1*(6), 1–13. https://link.springer.com/article/10.1007/s42979-020-00341-6

Feng, S. Y., Gangal, V., Wei, J., Chandar, S., Vosoughi, S., Mitamura, T., & Hovy, E. (2021). *A survey of data augmentation approaches for NLP*. arXiv preprint arXiv:2105.03075. https://doi.org/10.48550/arXiv.2105.03075

Gasparetto, A., Marcuzzo, M., Zangari, A., & Albarelli, A. (2022). A survey on text classification algorithms: From text to predictions. *Information*, *13*(2), 83. https://doi.org/10.3390/info13020083

Gillioz, A., Casas, J., Mugellini, E., & Abou Khaled, O. (2020, September). Overview of the transformer-based models for NLP tasks. *In 2020 15th Conference on Computer Science and Information Systems (FedCSIS)* (pp. 179–183). IEEE.

Golub, G. H., & Reinsch, C. (1971). Singular value decomposition and least squares solutions. *In Handbook for automatic computation: Volume II: Linear algebra* (pp. 134–151). Springer Berlin Heidelberg. https://doi.org/10.15439/2020F20

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. Cambridge: MIT Press.

Gopalakrishnan, S., Garbayo, L., & Zadrozny, W. (2025). Causality extraction from medical text using large language models (LLMs). *Information*, *16*(1), 13. https://doi.org/10.3390/info16010013

Grefenstette, G., & Tapanainen, P. (1994). What is a word, what is a sentence? Problems of tokenisation. COLING 1994 Volume 2: *The 15th International Conference on Computational Linguistics*, 2, 1038–1042.

Han, D., Lee, S., Song, M., & Cho, J. S. (2021). Change detection in unmanned aerial vehicle images for progress monitoring of road construction. *Buildings*, *11*(4), 150. https://doi.org/10.3390/buildings11040150

Han, J., Pei, J., & Kamber, M. (2011). *Data mining: Concepts and techniques*. Elsevier Science.

Han, W., Jiang, Y., Ng, H. T., & Tu, K. (2020). A survey of unsupervised dependency parsing. arXiv preprint arXiv: 2010.01535. https://doi.org/10.48550/arXiv.2010.01535

Harris, Z. S. (1954). Distributional structure. *WORD, 10*(2–3), 146–162.

Hassani, H., Silva, E. S., Unger, S., TajMazinani, M., & Feely, S. M. (2020). Artificial intelligence (AI) or intelligence augmentation (IA): What is the future? *AI*, *1*(2), 143–155. https://doi.org/10.3390/ai1020008

Htun, T. M., & Tun, Z. (2018). Algorithm tuning from comparative analysis of classification algorithms. *International Journal of Scientific and Research Publications (IJSRP), 8*(5), 2250–3153.

Johnson, S. J., Murty, M. R., & Navakanth, I. (2024). A detailed review on word embedding techniques with emphasis on Word2Vec. *Multimedia Tools and Applications*, *83*(13), 37979–38007. https://link.springer.com/article/10.1007/s11042-023-17007-z

Kaelbling, L. P., Littman, M. L., & Moore, A. W. (1996). Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, *4*, 237–285.

Karakoç, E., & Yılmaz, B. (2019, April). Deep learning based abstractive Turkish news summarization. *In 2019 27th Signal Processing and Communications Applications Conference (SIU)* (pp. 1–4). IEEE. https://doi.org/10.1109/SIU.2019.8806510

Kartal, Y. S., & Kutlu, M. (2020). Türkçe haber metinleri için makine öğrenmesi temelli özetleme. *In 2020 28th Signal Processing and Communications Applications Conference (SIU)* (pp. 1–4). IEEE. https://ieeexplore.ieee.org/document/9302096

Kanrar, S., Giri, N. C., Adak, D., Paul, S., Ghosh, S., & Das, S. (2023). Lstm models: A comprehensive analysis and applications. *Advancement in Image Processing and Pattern Recognition, 6(1),* 44-53. https://zenodo.org/records/7861337

Keneshloo, Y., Shi, T., Ramakrishnan, N., & Reddy, C. K. (2019). Deep reinforcement learning for sequence-to-sequence models. *IEEE Transactions on Neural Networks and Learning Systems*, *31*(7), 2469–2489. https://doi.org/10.1109/TNNLS.2019.2929141

Keraghel, I., Morbieu, S., & Nadif, M. (2024). A survey on recent advances in named entity recognition. ArXiv preprint arXiv: 2401.10825. https://doi.org/10.48550/arXiv.2401.10825

Khakurel, J., Penzenstadler, B., Porras, J., Knutas, A., & Zhang, W. (2018). The rise of artificial intelligence under the lens of sustainability. *Technologies*, *6*(4), 100. https://doi.org/10.3390/technologies6040100

Khalil, M., Said, M. N. H. M., Ibrahim, A. K., Gamal, M., & Mobarak, M. A. (2021). Tracking comets and asteroids using machine learning and deep learning: A review. *International Journal of Advanced Astronomy*, *9*(1), 26–27.

Khurana, D., Koli, A., Khatter, K., & Singh, S. (2023). Natural language processing: State of the art, current trends and challenges. *Multimedia Tools and Applications*, *82*(3), 3713–3744. https://link.springer.com/article/10.1007/s11042-022-13428-4

Koroteev, M. V. (2021). BERT: a review of applications in natural language processing and understanding. *arXiv preprint arXiv:2103.11943*. https://doi.org/10.48550/arXiv.2103.11943

Kuang, P., Cao, W., & Wu, Q. (2014). Preview on structures and algorithms of deep learning. *In 2014 11th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)* (pp. 176–179). IEEE. https://doi.org/10.1109/ICCWAMTIP.2014.7073385

Kublik, S., & Saboo, S. (2022). *GPT-3*. O'Reilly Media, Incorporated.

Kudo, T. (2018). Subword regularization: Improving neural network translation models with multiple subword candidates. *In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)* (Vol. 1, pp. 66–75). Association for Computational Linguistics. https://doi.org/10.48550/arXiv.1804.10959

Lai, S., Liu, K., He, S., & Zhao, J. (2016). How to generate a good word embedding. *IEEE Intelligent Systems*, *31*(6), 5–14. https://doi.org/10.1109/MIS.2016.45

LeCun, Y., Bengio, Y., & Hinton, G. E. (2015). Deep learning. *Nature*, *521*(7553), 436–444. https://www.nature.com/articles/nature14539

McDonald, S., & Ramscar, M. (2001). Testing the distributional hypothesis: The influence of context on judgements of semantic similarity. *In Proceedings of the Annual Meeting of the Cognitive Science Society* (Vol. 23, No. 23, pp. 611–616). https://escholarship.org/uc/item/6959p7b0

Mikolov, T., Le, Q. V., & Sutskever, I. (2013). Exploiting similarities among languages for machine translation. ArXiv preprint arXiv: 1309.4168. https://doi.org/10.48550/arXiv.1309.4168

Mohammed, M., Khan, M. B., & Bashier, E. B. M. (2016). *Machine learning: Algorithms and applications*. CRC Press. https://doi.org/10.1201/9781315371658

Nabi, W., & Xu, B. (2021). Applications of artificial intelligence and machine learning approaches in echocardiography. *Echocardiography, 38*(6), 982–992. https://doi.org/10.1111/echo.15048

Napte, K., & Mahajan, A. (2022). Deep learning based liver segmentation: A review. *Revue d'Intelligence Artificielle, 36*(6), 979–984. https://doi.org/10.18280/ria.360620

Nassif, A. B., Shahin, I., Attili, I. B., Azzeh, M., & Shaalan, K. (2019). Speech recognition using deep neural networks: A systematic review. *IEEE Access, 7*, 19143–19165. https://doi.org/10.1109/ACCESS.2019.2896880

Nozari, H., Szmelter-Jarosz, A., & Ghahremani-Nahr, J. (2022). Analysis of the challenges of artificial intelligence of things (AIoT) for the smart supply chain (case study: FMCG industries). *Sensors, 22*(8), 2931. https://doi.org/10.3390/s22082931

Radford, A. (2018). *Improving language understanding with unsupervised learning*. OpenAI. https://openai.com/research/language-unsupervised

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research, 21*(140), 1–67. https://www.jmlr.org/papers/v21/20-074.html

Rahman, A., Bowlin, G., Mohanty, B., & McGunigal, S. (2024). *Towards linguistically-aware and language-independent tokenization for large language models (LLMs)*. arXiv.

Ravì, D., Wong, C., Deligianni, F., Berthelot, M., Andreu-Pérez, J., Lo, B., & Yang, G.-Z. (2017). Deep learning for health informatics. *IEEE Journal of Biomedical and Health Informatics, 21*(1), 4–21. https://ieeexplore.ieee.org/document/7801947

Reimers, N., & Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. arXiv preprint arXiv:1908.10084. https://doi.org/10.48550/arXiv.1908.10084

Rumelhart, D. E., Widrow, B., & Lehr, M. A. (1994). The basic ideas in neural networks. *Communications of the ACM, 37*(3), 87–93.

Russell, S. J., & Norvig, P. (2010). *Artificial intelligence: A modern approach*. Pearson Education.

Sarker, I. H. (2021). Machine learning: Algorithms, real-world applications and research directions. *SN Computer Science*, *2*(3), 160. https://doi.org/10.1007/s42979-021-00592-x

Saufi, M. M., Zamanhuri, M. A. B., Mohammad, N., & İbrahim, Z. B. (2018). Deep learning for Roman handwritten character recognition. *Indonesian Journal of Electrical Engineering and Computer Science, 12*(2), 455–460. https://doi.org/10.11591/ijeecs.v12.i2.pp455-460

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*, *61*, 85–117. https://doi.org/10.1016/j.neunet.2014.09.003

Schneider, E. T. R., de Souza, J. V. A., Gumiel, Y. B., Moro, C., & Paraiso, E. C. (2021, June). A GPT-2 language model for biomedical texts in Portuguese. *In 2021 IEEE 34th International Symposium on Computer-Based Medical Systems (CBMS)* (pp. 474–479). IEEE. https://doi.org/10.1109/CBMS52027.2021.00056

Shabbir, J., & Anwer, T. (2018). Artificial intelligence and its role in near future. arXiv. https://doi.org/10.48550/arXiv.1804.01396

Shan, S., & Liu, Y. (2021). Blended teaching design of college students' mental health education course based on artificial intelligence flipped class. *Mathematical Problems in Engineering, 2021*, 1–10. https://doi.org/10.1155/2021/6679732

Shen, D., Wu, G., & Suk, H.-I. (2017). Deep learning in medical image analysis. *Annual Review of Biomedical Engineering, 19*(1), 221–248. https://doi.org/10.1146/annurev-bioeng-071516-044442

Song, X., Salcianu, A., Song, Y., Dopson, D., & Zhou, D. (2020). Fast WordPiece tokenization. arXiv. https://doi.org/10.48550/arXiv.2012.15524

Su, B., Ju, Y., & Dai, L. (2022). Deep learning for video application in cooperative vehicle-infrastructure system: A comprehensive survey. *Applied Sciences, 12*(12), 6283.

Sugiyama, H. (2019). Empirical feature analysis for dialogue breakdown detection. *Computer Speech & Language*, 54, 140-150. https://doi.org/10.1016/j.csl.2018.09.007

Sun, C., Qiu, X., Xu, Y., & Huang, X. (2019). *How to fine-tune BERT for text classification? In Lecture notes in computer science* (pp. 194–206). Springer. https://link.springer.com/chapter/10.1007/978-3-030-32381-3_16

Sun, Y., Wang, S., Li, Y., Feng, S., Chen, X., Zhang, H., Tian, X., Zhu, D., Tian, H., & Wu, H. (2019). ERNIE: Enhanced representation through knowledge integration. arXiv.

Sutskever, I. (2014). Sequence to sequence learning with neural networks. arXiv. https://doi.org/10.48550/arXiv.1904.09223

Şakar, T., & Emekci, H. (2025). Maximizing RAG efficiency: A comparative analysis of RAG methods. *Natural Language Processing, 31(1),* 1-25. https://doi.org/10.1017/nlp.2024.53

Tay, Y., Bahri, D., Metzler, D., Juan, D.-C., Zhao, Z., & Zheng, C. (2021, July). Synthesizer: Rethinking self-attention for transformer models. *In International Conference on Machine Learning* (pp. 10183–10192). PMLR. https://proceedings.mlr.press/v139/tay21a.html

TensorFlow. (2024). *Embedding projector [Interactive visualization]*. TensorFlow. https://projector.tensorflow.org

Güngör, M. (2023). *Wikipedia TR Summarization Dataset* [Veri kümesi]. Hugging Face. https://huggingface.co/datasets/musabg/wikipedia-tr-summarization

Thomas, C. (2019). Recurrent neural networks and natural language processing. *Towards Data Science*. https://towardsdatascience.com/recurrent-neural-networks-and-natural-language-processing-73af640c2aa1

Tian, B., Tian, F., & Xu, M. (2022). Research on innovation of college classroom instructional strategy from the perspective of deep learning. *Advances in Education, Humanities and Social Science Research, 2*(1), 535–540. https://doi.org/10.56028/aehssr.2.1.535

Topol, E. J. (2019). High-performance medicine: The convergence of human and artificial intelligence. *Nature Medicine, 25*(1), 44–56. https://www.nature.com/articles/s41591-018-0300-7

Unite.AI. (2025). *Machine Learning vs. Deep Learning*: Key Differences. Retrieved from https://www.unite.ai/machine-learning-vs-deep-learning-key-differences/

Upadhyay, A. (2017). Significant enhancements in machine translation by various deep learning approaches. *American Journal of Computer Science and Information Technology, 5*(2). https://hal.science/hal-03626992/

Wahl, B., Cossy-Gantner, A., Germann, S., & Schwalbe, N. R. (2018). Artificial intelligence (AI) and global health: How can AI contribute to health in resource-poor settings? *BMJ Global Health, 3*(4). https://doi.org/10.1136/bmjgh-2018-000798

Walton, D. (1996). A pragmatic synthesis. In Fallacies arising from ambiguity (pp. 1–20). Springer.

Wang, L., & Fu, S. (2021). Container multimodal cooperative transportation management information system based on artificial intelligence technology. *Mathematical Problems in Engineering, 2021*, 1–14. https://doi.org/10.1155/2021/1272221

Wang, S., Luo, J., & Luo, L. (2022). Large-scale text multiclass classification using Spark ML packages. *Journal of Physics: Conference Series, 2171(1)*. https://doi.org/10.1088/1742-6596/2171/1/012022

Wang, Y., Ge, X., Ma, H., Qi, S., Zhang, G., & Yao, Y. (2021). Deep learning in medical ultrasound image analysis: A review. *IEEE Access, 9*, 54310–54324. https://doi.org/10.1109/ACCESS.2021.3071301

Yasrab, R., Pound, M. P., French, A. P., & Pridmore, T. P. (2020). PhenomNet: Bridging phenotype-genotype gap: A CNN-LSTM based automatic plant root anatomization system. bioRxiv. https://doi.org/10.1101/2020.05.03.075184

Young, T., Hazarika, D., Poria, S., & Wang, Z. (2018). Recent trends in deep learning based natural language processing [Review article]. *IEEE Computational Intelligence Magazine, 13*(3), 55–75. https://doi.org/10.1109/MCI.2018.2840738

Yousuf, H., Lahzi, M., Salloum, S. A., & Shaalan, K. (2021). A systematic review on sequence-to-sequence learning with neural network and its models. *International Journal of Electrical & Computer Engineering, 11*(3), 2088–8708. https://doi.org/10.11591/ijece.v11i3.pp2315-2326

Yu, Y., Li, M., Liu, L., Li, Y., & Wang, J. (2019). Clinical big data and deep learning: Applications, challenges, and future outlooks. *Big Data Mining and Analytics, 2*(4), 288–305. https://doi.org/10.26599/BDMA.2019.9020007

Zhang, S., Xie, X., & Yang, X. (2020). A brute-force black-box method to attack machine learning-based systems in cybersecurity. *IEEE Access, 8*, 128250–128263. https://doi.org/10.1109/ACCESS.2020.3008433

Zhou, P., & Feng, J. (2017). The landscape of deep learning algorithms. arXiv. https://doi.org/10.48550/arXiv.1705.07038

Ekler -1 Uygulama Kaynak Kodu

```
import pandas as pd
import numpy as np
from sentence_transformers import SentenceTransformer
from transformers import T5Tokenizer, T5ForConditionalGeneration
from sklearn.metrics.pairwise import cosine_similarity
from rouge_score import rouge_scorer
import os

model_name = "t5-large"
tokenizer_t5 = T5Tokenizer.from_pretrained(model_name)
model_t5 = T5ForConditionalGeneration.from_pretrained(model_name)
sentence_transformer_model = SentenceTransformer("paraphrase-MiniLM-L6-v2")
def save_to_excel_incremental(data, file_path)
    if os.path.exists(file_path):
        existing_data = pd.read_excel(file_path)
        new_data = pd.concat([existing_data, pd.DataFrame(data)], ignore_index=True)
    else:
        new_data = pd.DataFrame(data)
    new_data.to_excel(file_path, index=False)
def summarize_with_t5_large(text):
    input_text = "summarize: " + text
    inputs = tokenizer_t5(input_text, return_tensors="pt", max_length=512,
truncation=True)
    summary_ids = model_t5.generate(
        inputs.input_ids,
        max_length=200,
        min_length=50,
```

```
        length_penalty=2.0,
        num_beams=4,
        early_stopping=True
    )
    return tokenizer_t5.decode(summary_ids[0], skip_special_tokens=True)
def summarize_with_t5_and_sentence_transformer(text):
    sentences = text.split(". ")
    if len(sentences) == 1:
        return summarize_with_t5_large(text)
    embeddings = sentence_transformer_model.encode(sentences)
    similarity_matrix = cosine_similarity(embeddings)
    sentence_similarity_sums = similarity_matrix.sum(axis=1)
    important_indices = np.argsort(sentence_similarity_sums)[::-1]
    important_sentences = [sentences[i] for i in important_indices[:7]]
    summary_input = " ".join(important_sentences)
    return summarize_with_t5_large(summary_input)
def calculate_rouge(reference, summary):
    scorer = rouge_scorer.RougeScorer(["rouge1", "rouge2", "rougeL"],
use_stemmer=True)
    scores = scorer.score(reference, summary)
    return {
        "ROUGE-1": scores["rouge1"].fmeasure,
        "ROUGE-2": scores["rouge2"].fmeasure,
        "ROUGE-L": scores["rougeL"].fmeasure
    }
rouge_scores_data = []
file_path = '/content/drive/MyDrive/summary_rouge_results_wikisum.xlsx'
```

```
if os.path.exists(file_path):
    processed_data = pd.read_excel(file_path)
    processed_indices = processed_data.index
else:
    processed_indices = []
for i, text in enumerate(test_texts):
    if i in processed_indices:
        continue
    reference = test_summaries[i]
    summary_t5 = summarize_with_t5_large(text)
    summary_t5_with_st = summarize_with_t5_and_sentence_transformer(text)
    rouge_scores_t5 = calculate_rouge(reference, summary_t5)
    rouge_scores_t5_with_st = calculate_rouge(reference, summary_t5_with_st)
    rouge_scores_data.append({
        "text": text,
        "reference_summary": reference,
        "t5_summary": summary_t5,
        "t5_with_st_summary": summary_t5_with_st,
        "ROUGE-1 (T5)": rouge_scores_t5["ROUGE-1"],
        "ROUGE-2 (T5)": rouge_scores_t5["ROUGE-2"],
        "ROUGE-L (T5)": rouge_scores_t5["ROUGE-L"],
        "ROUGE-1 (T5 + ST)": rouge_scores_t5_with_st["ROUGE-1"],
        "ROUGE-2 (T5 + ST)": rouge_scores_t5_with_st["ROUGE-2"],
        "ROUGE-L (T5 + ST)": rouge_scores_t5_with_st["ROUGE-L"]
    })
    save_to_excel_incremental(rouge_scores_data, file_path
```

Özgeçmiş

Araştırma Alanları**:** Bilgi Teknolojileri, Hesaplamalı Bilimler, Derin Öğrenme, Doğal Dil İşleme

Eğitim

Doktora Yönetim Bilişim Sistemleri, İstanbul Topkapı Üniversitesi, 2025 / İstanbul / Türkiye
Yüksek Lisans (MBA) Greenwich Üniversitesi, 2009 / Londra / Birleşik Krallık
Lisans Bilgisayar Mühendisliği, Doğu Akdeniz Üniversitesi, 2004 / Gazimağusa / KKTC

İş Deneyimi

Kurucu Ortak, DeepMean Yapay Zekâ Ltd., (2023-Devam ediyor) / Samsun / Türkiye
Bilgi Teknolojileri Danışmanı, Koban Stratejik Bilgi Teknolojileri, (2011- Devam ediyor) / Samsun / Türkiye
İş Zekâsı ve Veri Ambarı Mühendisi, Türk Telekom, 2010-2011 / İstanbul / Türkiye
Bilgi Teknolojileri Uzmanı, Howard Gurpınar Şirketi, 2008-2010 / Londra / Birleşik Krallık
Yazılım Mühendisi, Turkcell, 2007-2008 / İstanbul / Türkiye

Tezler

Yüksek Lisans Tezi: Müşteri İlişkileri ve Veri Madenciliğinin Müşteri Sadakati Üzerindeki Etkisi, 2009
Doktora Tezi: Mevcut Derin Doğal Dil İşleme Modellerinin Türkçe İçin Performansının Artırılması, 2025