T.C.

YILDIZ TEKNİK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

MASAL KAHRAMANLARININ TESPİTİ VE ARALARINDAKİ DUYGUSAL İLİŞKİNİN ÇIKARILMASI

Samir GANBARLI

YÜKSEK LİSANS TEZİ

Bilgisayar Mühendisliği Anabilim Dalı

Bilgisayar Mühendisliği Programı

Danışman

Prof. Dr. Banu DİRİ

Temmuz, 2024

T.C.

YILDIZ TEKNİK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

MASAL KAHRAMANLARININ TESPİTİ VE ARALARINDAKİ DUYGUSAL İLİŞKİNİN ÇIKARILMASI

Samir GANBARLI tarafından hazırlanan tez çalışması 17.07.2024 tarihinde aşağıdaki jüri tarafından Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı, Bilgisayar Mühendisliği Programı **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Prof. Dr. Banu DİRİ
Yıldız Teknik Üniversitesi
Danışman

Jüri Üyeleri

Prof. Dr. Banu DİRİ, Danışman
Yıldız Teknik Üniversitesi _____

Dr. Öğr. Üyesi Furkan ÇAKMAK, Üye
Yıldız Teknik Üniversitesi _____

Dr. Öğr. Üyesi Öznur ŞENGEL, Üye
İstanbul Kültür Üniversitesi _____

Danışmanım Prof. Dr. Banu DİRİ sorumluluğunda tarafımca hazırlanan "Masal Kahramanlarının Tespiti ve Aralarındaki Duygusal İlişkinin Çıkarılması" başlıklı çalışmada veri toplama ve veri kullanımında gerekli yasal izinleri aldığımı, diğer kaynaklardan aldığım bilgileri ana metin ve referanslarda eksiksiz gösterdiğimi, araştırma verilerine ve sonuçlarına ilişkin çarpıtma ve/veya sahtecilik yapmadığımı, çalışmam süresince bilimsel araştırma ve etik ilkelerine uygun davrandığımı beyan ederim. Beyanımın aksinin ispatı halinde her türlü yasal sonucu kabul ederim.

Samir GANBARLI

İmza

Bütün eğitim hayatım boyunca beni destekleyen büyükanneme

TEŞEKKÜR

Tez çalışmamın gerçekleşmesinde benimle tecrübe ve bilgilerini paylaşan, projede kullanılan veri setlerinin bulunmasında yardımcı olan çok değerli danışman hocam Banu DİRİ'ye, çalışmam boyunca maddi ve manevi desteğini benden esirgemeyen değerli aileme teşekkür ederim.

Samir GANBARLI

İÇİNDEKİLER

SİMGE LİSTESİ vii
KISALTMA LİSTESİ viii
ŞEKİL LİSTESİ ix
TABLO LİSTESİ xi
ÖZET xii
ABSTRACT xiv
1 GİRİŞ 1
1.1 Giriş ... 1
1.2 Literatür Özeti ... 3
1.3 Hipotez ... 6
1.4 Tezin Ana Hatları ... 6
2 VERİ SETLERİ 7
2.1 Varlık İsmi Tanıma Modeli için Kullanılan Veri Seti ... 7
2.2 Duygu Analizi Modeli için Kullanılan Veri Seti ... 9
2.3 Veri İşleme ... 9
3 YÖNTEM 10
3.1 Python ... 10
3.2 Pytorch ... 10
3.3 Zeyrek (Zemberek) ... 11
3.4 Spacy ... 12
3.5 Metin Temsili Yöntemleri ... 14
3.5.1 Bag of Words ... 14
3.5.2 TF-IDF ... 15
3.5.3 One Hot Encoding ... 15
3.6 Tokenizasyon ... 16
3.7 Kelime Gömmeleri ... 17
3.8 Derin Öğrenme Teknikleri ... 19
3.8.1 RNN (Tekrarlayan sinir ağları), LSTM ... 19
3.8.2 Dönüştürücüler ... 21
4 SİSTEM TASARIMI 26
4.1 Varlık İsmi Tanıma Modeli ... 27
4.2 Duygu Analizi Modeli ... 30

4.3 Modellerin Birlikte Kullanımı 32
5 DENEYSEL SONUÇLAR **38**
6 SONUÇ **50**
KAYNAKÇA **52**
A MASAL METİNLERİ **54**
TEZDEN ÜRETİLMİŞ YAYINLAR **57**

SİMGE LİSTESİ

F	Forget Gate (LSTM)
h	Hidden Layer Output
I	Input Gate (LSTM)
O	Output Gate (LSTM)
W	Weight

KISALTMA LİSTESİ

BERT	Bidirectional Encoder Representation from Transformers
BOW	Bag of Words
CBOW	Continuous Bag of Words
Glove	Global Vectors for Words Representations
GPU	Graphics Processing Unit
IDF	Inverse Document Frequency
LSTM	Long-Short Term Memory
NER	Named Entity Recognition
TF	Term Frequency

ŞEKİL LİSTESİ

Şekil 2.1 Varlık ismi tanıma modelinin eğitim veri setinin ilk versiyonu.....8
Şekil 2.2 Varlık ismi tanıma modelinin veri setinin değişimden sonraki versiyonuna ait görüntü 8
Şekil 2.3 Duygu analizi modelinin veri setinin bir bölümüne ait görüntü 9
Şekil 3.1 Spacy kütüphanesinin yüklenmesine ait sözde kod 13
Şekil 3.2 Bağlılık ayrıştırma yöntemine ait bir örnek 14
Şekil 3.3 Çalışmada kullanılan bağlılık ayrıştırmasına ait sözde kod 14
Şekil 3.4 "BoW" yöntemi için örnek.....15
Şekil 3.5 BERT tokenizasyon aşamasına ait örnek bir görüntü 16
Şekil 3.6 Kelime gömme ile yaratılan çokboyutlu kelime uzayı ait görüntü 17
Şekil 3.7 CBOW yöntemine ait basit bir diagram 18
Şekil 3.8 Skip-gram modelina ait basit bir görüntü.....18
Şekil 3.9 BERT gömme katmanına ait bir görüntü [21].....19
Şekil 3.10 Tekrarlayan sinir ağının yapısına ait basit bir görsel [23].....20
Şekil 3.11 RNN yapısına ait basit bir görsel [24].....20
Şekil 3.12 LSTM ağının giriş, unutma ve çıktı geçitlerine ait görsel [25].....21
Şekil 3.13 Çok-kafalı dikkat mekanizmasına ait basit bir görsel [26].....23
Şekil 3.14 Dönüştürücü mimarisine ait görsel [13].....24
Şekil 3.15 BERT dönüştürücüsüne ait görsel.....25
Şekil 4.1 Uygulamanın algoritmik akışına ait basit bir görsel……...27
Şekil 4.2 Varlık ismi tanıma modelinin veri setindeki etiket sayısı bilgileri 28
Şekil 4.3 Varlık ismi tanıma modelinin sonuçlarından kişi isimlerinin çıkarılması 29
Şekil 4.4 Varlık ismi tanıma modelinin joblib ile kaydedilmesine ait kod 29
Şekil 4.5 Duygu analizi modelinin eğitim sürecinin adımlarına ait diagram 31
Şekil 4.6 Metin üzerinde yapılan bazı önişlemlerin akışı.....33
Şekil 4.7 Bağlılık ayrıştırması modelinin çıktılarının listeye eklenmesine ait sözde kod 34
Şekil 4.8 Listede tekrarlanan isimlerin silinmesinin basit akış diagramı 35
Şekil 4.9 Varlık ismi tanıma ve bağlılık ayrıştırması modellerinin çıktı listelerinin birleştirilmesine ait akış diagramı.....36

Şekil 4.10 İki boyutlu karakter listesine ait görüntü.....37
Şekil 4.11 Duygu analizi modelinin uygulanmasının akışı 37
Şekil 5.1 Masal kahramanları arasındaki duygusal ilişkinin sayısı……. 45
Şekil 5.2 Balıkçı masalının tamamına ait ilişki diagramı 45
Şekil 5.3 Çirkin Prens masalının tamamına ait ilişki diagramı 46
Şekil 5.4 Kınalı Kuzu masalının tamamına ait ilişki diagramı.....46
Şekil 5.5 Kınalı Kuzu masalının giriş bölümündeki duyguya ait ilişki diagramı..48
Şekil 5.6 Kınalı Kuzu masalının gelişme bölümünde duyguya ait ilişki diagramı49

TABLO LİSTESİ

Tablo 5.1 Test verisinde yer alan masallar ve sahip olduğu özellikler 38
Tablo 5.2 Varlık ismi tanıma modeli ilk iki karakter tespiti 40
Tablo 5.3 Varlık ismi tanıma modeli diğer karakterler tespiti 42
Tablo 5.4 Masallardaki karakterlerin çıkarım başarısı.....43
Tablo 5.5 Duygu analizi modelinin başarısı 46

ÖZET

Masal Kahramanlarının Tespiti ve Aralarındaki Duygusal İlişkinin Çıkarılması

Samir GANBARLI

Bilgisayar Mühendisliği Anabilim Dalı
Bilgisayar Mühendisliği Programı
Yüksek Lisans Tezi

Danışman: Prof. Dr. Banu DİRİ

Günümüzde yapay zekanın bir alt dalı olan doğal dil işleme otomatik makine çevirisi, duygu analizi, epostalardan spam tespiti, sosyal medya analizi gibi pratik alanlarda sıkça kullanılmaktadır. Yapay zekanın bu branşında yapılan araştırmalar ve geliştirilen modeller ile bilgisayarların metinleri anlama, onları analiz etme, bilgi çıkarma ve soru cevaplama gibi önemli işlemleri yerine getirme becerileri de gelişmiştir. Makine öğrenmesi, derin öğrenme modelleri ve son yıllarda dönüştürücüler ile doğal dil işlemenin bir çok uygulama alanında büyük başarılar alınmıştır. Masallardan ana karakterlerin çıkarılması ve karakterler arasındaki duygusal ilişkilerin belirlenerek bir sosyal ağ haritası yardımıyla gösterilmesi de doğal dil işlemedeki güncel konulardan biridir. Yapılan tez çalışmasında bu konu araştırılmış ve üç aşamada Türkçe yazılmış masallardaki karakterlerin tespit edilmesi ve aralarındaki duygusal ilişkinin çıkarılması için bir uygulama geliştirilmiştir. Kullanılan modellerden ilk ikisi kişi olarak isimlendirilmiş varlıkların çıkarılması ve bağımlılık ayrıştırma analizi yapılarak masaldaki ana karakterlerin tespit edilmesi olmuştur. Üçüncü aşamada ise tespit edilen karakterlerin geçtiği cümleler çıkarılarak, karakterler arasındaki duygusal ilişkinin derecesi belirlenmiştir. Son aşamada ise ilişkiler durumlarına göre sosyal bir ağ yardımıyla görselleştirilmiştir.

Anahtar Kelimeler: Karakterlerin tespiti, bağlılık ayrıştırma, RNN, LSTM, duygu analizi.

ABSTRACT

Identification of Fable Heroes and The Emotional Relationship between Them

Samir GANBARLI

Department of Computer Engineering

Master of Science Thesis

Supervisor: Prof. Dr. Banu DIRI

Today, natural language processing (NLP), a subfield of artificial intelligence, is frequently applied in practical areas such as machine translation, sentiment analysis, spam detection in emails and social media analysis. Research and model development in this branch of artificial intelligence have enhanced computers' abilities to understand texts, analyze them, extract information, and perform tasks such as question answering. Significant achievements have been made in various applications of natural language processing through machine learning, deep learning models such as, RNN or LSTM and more recently, transformers. Extracting main characters from fables, determining the emotional relationships between characters and displaying them in the form of a social network map is one of the current topics in natural language processing. In the thesis study, this issue was researched, and an application was developed in three stages to identify the characters in the fable written in Turkish and to extract the emotional relationships between them. The first two of the models involved named entity recognition and performing dependency parsing analysis to identify the main characters in the fable. In the third stage, sentences containing the identified characters were extracted, and the sentimental relationship between the characters was determined. Finally, sentimental relationships have been visualized in the form of a social network based on their statuses.

Keywords: Character identification, dependency parsing, RNN, LSTM, sentiment analysis.

1
GİRİŞ

1.1 Giriş

İlk çağlardan beri insanlar kendilerini ifade etmek ve anlaşılabilmek için farklı yollar denemişlerdir. Toplumsal gelişmeler zaman içerisinde değişik anlaşma yollarının benimsenmesine neden olmuştur. Duman ile anlaşma, işaret diliyle anlaşma, duvarlara çizilen görsel figürlerle anlaşmak zamanla yerini yazarak anlaşmaya bırakmıştır. İlerleyen zaman diliminde yazının da daha sistematik bir şekilde kullanımı ve gelişimi ile edebiyat doğmuştur. Kurgu veya gerçek olaylardan esinlenilerek duygu ve düşüncelerin estetik bir şekilde anlatılması ile güzel sanatların bir dalı olan edebiyatın altında yer alan roman karşımıza çıkmıştır.

Edebi eserler, insanların düşüncelerini ve hislerini büyük ölçüde şekillendirebilen önemli evrensel bir araçtır. Roman veya hikayelerin okuyucular üzerinde bıraktıkları etkiler sosyoloji, psikoloji, kültür veya dil bilimi araştırmacıları tarafından incelenip, analiz sonrasında alınan sonuçlar ile başka araştırmalara girdi olarak kullanılmaktadır. Roman veya hikayedeki ana karakterlerin tespiti, karakterler arasındaki etkileşimin duygu durumu özellikle doğal dil işleme alanında çalışanlar için bir araştırma konusu olmuştur. Karakterlerin tespit edilmesi, aralarındaki ilişkinin çıkarılması ve duygu durumunun tespit edilmesinden sonra hikayenin anlamsal derinliğine inip analiz yapılabilmesi mümkün olmaktadır.

Doğal dil işlemenin farklı alt alanlarındaki gelişmeler, metinlerin analizini ve anlaşılırlığını kolaylaştırmaktadır. Özellikle, bu alanlarda dikkat mekanizması ile çalışan dönüştürücülerin kullanılması sonucu elde edilen gelişmeler oldukça dikkat çekicidir. Genel olarak, metin analizi aşamasında varlık isimlerinin tespiti, duygu analizi, metin özetleme veya metin sınıflandırma gibi konular için geliştirilmiş olan bu dil modelleri kullanılmaktadır. Bu modellerin bireysel veya kurumsal olarak doğrudan veya ince ayar yapılarak kullanılması ve yüksek başarı sonuçların alınması bu alanda yeni modellerin çıkarılmasına ve bu modellerinde çalışmalarda yaygın olarak kullanılmasına neden olmaktadır. Örneğin, bir duygu analizinin müşteri geri bildirimlerinde, sosyal medya analizlerinde kullanılması bir hikaye analizinde hikayedeki kahramanlar arasındaki ilişkinin duygusunun çıkarılmasında

da kullanılması söz konusudur. Aynı durum, varlık isimlerinin tespit edilmesi için de geçerlidir. Örneğin müşteri geri bildirimleri analiz edilirken ürün ve müşteri isimlerinin çıkarılması önemli iken bir hikayenin analizinde hikaye kahramanlarının tespitinde önemli olmaktadır.

Bu tez kapsamında Türkçe dilinde yazılmış masallardaki karakterlerin tespit edilerek çıkarılması ve karakterler arasındaki ilişkinin masalın başından sonuna kadar olan süreçte duygusal anlamda olumlu veya olumsuz olarak değişiminin tespit edilmesi yapılmaktadır. Bu projenin çıktısı olacak olan üründen edebiyat alanında çalışan öğrenci veya araştırmacılar tarafından edebi bir eserin, haber metninin incelenmesinde veya sosyal bilimler ile ilgili araştırmalarda tarihi veya sosyolojik içerikli metinlerin analiz edilmesinde yararlanılabilir. Bu çalışmada masaldaki karakterlerin tespiti ve aralarındaki ilişkinin duygusal eğiliminin çıkarılmasında üç derin öğrenme modeli kullanılmıştır. Geliştirilen uygulamanın ilk aşamasında girdi olarak bir masal verilir. Sonrasında, verilen metin içerisindeki karakterlerin tespit edilmesi için türkçe veri seti ile eğitilen varlık ismi tespiti modeli ve bağlılık ayrıştırması çalıştırılır. Kullanılan varlık ismi tespit modeli 60 bin cümleden oluşan bir veri setiyle eğitilmiş, dikkat mekanizması ile çalışan BERT dönüştürücüsüdür. Varlık ismi tanıma modelleri, cümlelerdeki özel isimlerin – kişi, yer, organizasyon, tarih gibi varlıkların belirlenmesi için kullanılır. Bu modellerin çıktı verileri genelde cümledeki isimler ve bu isimlerin etiketidir. Proje kapsamında sadece “B-PERSON” ve “I-PERSON” ile etiketli olan isimler kullanılmıştır. Bu model masal karakterleri olan isimlerin ilk harfinin büyük yazıldığı pek çok metin için faydalı olsa da, bazı masallar için iyi sonuçlar vermemektedir. Bunun sebebi bazı masallarda karakterlerin varlığı özel isimler yerine küçük harfle yazılmış genel isimlerle belirtiliyor olmasıdır. Örneğin, “Kırmızı Başlıklı Kız” masalında karakterlerin adları “kız”,” büyükanne”,” kurt” gibi genel isimlerle ifade edilmektedir. Böyle durumlarda yukarıda anlatıldığı gibi “Varlık İsmi Tespit” modeli yerine cümleler üzerinde biçimbirimsel (morphological) analiz yapabilen ve özneleri tespit edebilen bağlılık ayrıştırması yöntemi kullanılır. Bu yöntemde kullanılan kütüphane çıktı olarak cümledeki kelimelere ilgili etiketleri atar, kelimeler arasındaki ilişkileri çıkarır ve biz de sadece özneyi bildiren etiketli kelimelere odaklanırız. Sonraki aşamada karakterler arasındaki duygusal ilişkiyi bulmak adına karakterler ikili ikili seçilip, iki karakterin geçtiği cümleler tespit

edilir. Yine Türkçe için duygu analizi derin öğrenme modeli kullanılarak bu cümlelerdeki duygusal eğilim bulunur.

1.2 Literatür Özeti

Literatür taraması kapsamında yazılı metinler içerisinden tespit edilen karakterler ve aralarındaki duygusal ilişkinin çıkarılması üzerine, çoğunlukla İngilizce için yapılan çalışmalar ele alınmıştır.

Matt Fernandez vd. [1] hikayede bulunan Antagonist ve Protagonist karakterlerin tespit edilmesi için birkaç aşamadan oluşan bir yaklaşım önermişlerdir. İngilizce metinler için önerilen bu yaklaşımda ilk olarak karakterlerin tespiti için "Stanford NLP" kütüphanesi ile yazılmış varlık ismi tespiti modelini kullanmışlardır. Bu modeli çalıştırmadan önce bazı cümlelerde isimlerin yerine kullanılan zamirlerin karşılığı olan isimleri yazmak için zamir çözümlemenin kullanılması seçilmiştir. Son aşamada ise aralarındaki duygusal tonun belirleneceği iki karakter seçilip, iki ismin birlikte geçtiği cümleler seçilmiş ve "SentiWordNet" kütüphanesi kullanılarak bu cümlelerin pozitif veya negatif duygunun hangisine sahip olduğu tespit edilmiştir.

Devisree V ve P.C. Reghu Raj [2], hibrid olarak gözetimli ve gözetimsiz öğrenme kullanarak hikayedeki ana karakterler arasındaki anlamsal ilişkiyi belirleyen bir metot geliştirmişlerdir. İki karakter arasındaki ilişkinin romantik, dostluk veya ailevi türden olduğunu belirleyen bu makine öğrenmesi uygulamasında özdeşlik çözümleme kullanıldıktan sonra "Stanford Named Entity Recognizer" modeli ile metindeki karakterler belirlenmiştir. Anlamsal ilişkinin belirlenmesi için denetimli öğrenme modeli olarak "Naive Bayes", diğer model için ise "WordNet" ile çalışan "UMBC semantic text service" kullanılmıştır.

Briane Paul Samson ve Ethel Ong [3], doğal dil işlemede sıkca kullanılan GATE ve ConceptNet araçlarını kullanarak hikayedeki semantik ilişkilerin belirlenmesi için bir yaklaşım öne sürmüşlerdir. Araştırmada iki isim arasındaki varlığın türü ("Paris", "şehir"), cismin hangi maddeden yapıldığı ("ev", "ahşap") ve varlığın becerileri ("aslan", "avlanmak") tipli bağlantıları tespit etmek için doğal dil işlemede sıkça kullanılan ve birçok dile destek veren ConceptNet kütüphanesi veya

anlamsal ilişki ağı kullanılmıştır. Metinde ön işleme, anotasyon veya metin parçası etiketleme için ise GATE aracı kullanılmıştır.

Burhan Akkuş [4] masal veya hikayede ana karakterlerin tespit edilip, aralarındaki duygusal eğilimin doğal dil işleme araçları ile tespit edilmesini araştırmıştır. İlk olarak karakter tespiti aşaması için fiil analizi ve bağlılık ayrıştırması uygulamıştır. İlave olarak, ismin kendisi yerine kullanılan zamirlerin isimle yer değiştirmesi için "Stanford NLP" kütüphanesi ile özdeşlik çözümlemeden yararlanmıştır. Bir sonraki adımda karakterlerin birlikte geçtiği cümleler seçilmiş ve "NLTK Sentiment Intensity Analyzer" ile duygu analizi yapılmıştır.

Pierre-Yves vd. [5] gözetimsiz öğrenme yolu ile metinlerdeki "bornIn"," marriedTo"," locatedIn" tipli bağlantıların bulunması için "PromptORE – Prompt Based Open Relation Extraction" isimli modeli geliştirmişlerdir. Gözetimsiz öğrenme ile büyük hacimli etiketlenmiş verinin eğitimine gerek duyulmaması ve hiperparametlerin sayısının minimuma indirilmesi araştırmanın esas odak noktaları olmuştur. İlk olarak metindeki cümleler BERT modeli ile matematiksel vektör olarak gösterilip, "PromptORE" modeli ile benzer anlamsal ilişki içeren cümleler kümelenmiştir. Cümleler BERT ile vektöre dönüştürüldükten sonra Öklid uzaklığı ile normalizasyon işlemi uygulanmıştır.

Cuong Xuan Chu vd. [6] araştırmalarında hikaye, masal gibi kurgusal metinlerden ana karakterler arasındaki dostluk veya düşmanlık ilişkilerini belirlemek için karakterler tespit edildikten sonra, BERT ile eğitilmiş bir çok bağlamlı model kullanılmasını önermişlerdir. Bu model ile ilk aşamada metindeki karakterlerin birlikte kullanıldığı cümlelerin önem derecesi belirlenip, daha büyük öncelik taşıyan cümleler incelenerek ilişkiler belirlenmiştir. Karakterlerin kendilerinin anlamsal bilgisine erişmek için ise "Spacy" kullanılmıştır.

Shadi Shahsavari vd. [7] film veya romandaki ana karakterler arasındaki anlamsal bağı belirlemek için kullanıcılar tarafından yapılan değerlendirme ve yorumları inceleyip sonuç veren bir model geliştirmişlerdir. Özellikle meşhur veya başarılı roman veya filmlere çok fazla sayıda yorum geldiği için bu yorumlardan bağlılık ayrıştırma yoluyla esas karakterleri tespit edip, denetimsiz öğrenme yöntemi ile ilişkiyi belirlemişlerdir. Cümleleri matematiksel olarak ifade etmek için "BERT Embedding" ile vektöre dönüştürmüş ve Öklid mesafe yöntemini kullanmışlardır.

Krishna Janakiraman [8] “Stanford NLP” nin varlık ismi tanıma aracını kullanarak hikaye içinden ana karakterleri tespit edip, denetimsiz öğrenme yöntemi ile aralarındaki dostluk, düşmanlık, aile, romantik, kardeşlik gibi ilişki türlerini belirlemişlerdir. Kümeleme yönteminden elde edilen anlamsal bağ skoru ile yukarıda belirtilen ilişki türlerinden hangisine daha yakın olduğu tespit edilmiştir.

Hardik Vala vd. [9] metinlerde karakterlerin belirlenmesinde yaşanan bazı zorlukları önlemek için birkaç aşamadan oluşan bir yöntem önermişlerdir. “Stanford CoreNLP” kütüphanesinin varlık ismi tanıma ve özdeşlik çözümleme modellerini kullanarak elde edilen karakter isimlerini graf veri yapısı ile ifade etmişlerdir. Bu grafta isimler düğümleri, düğümler arasındaki bağlantılar da ilişkileri göstermektedir. Metinde bir karakterin farklı cümlelerde farklı sözlerle ifade edildiği tespit edildiğinde ikinci veya sonra gelen isimlerin yerine ilk olan yazılıp karakter tespitinin sonucu kesinleştirilmiştir.

Desmond C. Ong [10] metinlerde mutluluk, üzüntü, hayal kırıklığı gibi duygusal eğilimi olan cümleleri tespit edip, eğilimin sebebini belirleyen bir uygulama geliştirmiştir. Bağlılık ayrıştırması ve SVM makine öğrenmesi modeli kullanılarak geliştirilen uygulamada girdi olarak metin, metinde duygusal eğilim taşıyan paragraf ve bu eğilimin sebebi verilmiştir.

Despina Christou ve Grigorios Tsoumakas [11] 19. yüzyılda yazılmış Yunanca metinlerden iki öğe arasındaki “workAt”, “artHero” gibi 6 farklı anlamsal bağlantıyı belirlemek için “RedSandTLit” isimli, dönüştürücü tabanlı bir model geliştirmişlerdir. Eğitim için veri seti uzaktan gözetim yöntemi ile otomatik olarak hazırlanmıştır. Modelin giriş yapısı metin, öğelerin isimleri ve aralarındaki ilişki olarak kurulmuştur.

Alisha Bajracharya vd. [12] kural tabanlı yaklaşım ile hikayedeki esas karakterler arasındaki aile, dostluk gibi bağlantıları tespit eden bir uygulama geliştirmişlerdir. Özyinelemeli bir algoritma ile cümleler “NLTK” kütüphanesinin sunduğu tokenlaştırma ve metin parçası etiketleme gibi metotlarla ön işlemeden geçtikten sonra ağaç veri yapısında tanımlanmıştır. Sonrasında, cümlelerin ağaçdaki, en sonuncu seviyedeki yapraklarda kalan kısımlarında fiil veya isim analizi yapılıp karakterler ve aralarındaki ilişki belirlenmiştir.

Nijila M. ve Kala M.T [13] evrişimsel sinir ağlarını kullanarak esas karakterlerin tespiti yapıldıktan sonra aralarındaki aile, dostluk, romantik ilişki gibi bağlantıları belirleyen bir model geliştirmişlerdir. İlk olarak doğal dil işlemede uygulanan bazı klasik ön işlemlerden sonra, "Stanford Deterministic Coreference Resolution System" kütüphanesi ile özdeşlik çözümlemeyi kullanarak, karakterleri ifade eden zamirler orjinal isimleri ile değiştirilip, iki karakterin birlikte geçtiği cümleler seçilmiştir. Son aşamada ise doğal dil işlemede sıkca kullanılan, matris veri yapısı ile çalışan evrişimsel sinir ağları modeli ile ilişkiler tespit edilmiştir. Cümle gömme işlemi için de "Word2Vec" kullanılmıştır.

1.3 Hipotez

Doğal dil işleme araçları ve yöntemlerinin hızla gelişmesi sayesinde metinlerin otomatik olarak analizi daha hızlı ve verimli bir şekilde yapılmaya başlamıştır. Yapay zekanın bu branşının en çok geliştirilen alanları içerisinde duygu analizi ve varlık ismi tespiti gelmektedir. Bu araştırma alanlarında geliştirilen modellerin birlikte kullanılması sayesinde de yeni uygulamalar geliştirilmektedir. Metin içinde geçen karakterler arasındaki duygusal ilişkinin belirlenmesi de roman, hikaye, haber yazıları veya bilimsel yazıların incelemesi aşamasında araştırmacılar tarafından araç olarak kullanılmaktadır. Kullanıcılar bu uygulamalar ile bir yazıyı girdi olarak verdiklerinde, yazı veya masaldaki ana karakterlerin listesini çıkarabilir ve seçilen karakterler arasındaki duygusal ilişkiyi tahmin edebilir.

1.4 Tezin Ana Hatları

Tez çalışmasının ikinci bölümünde, uygulamanın parçası olan yapay zeka modellerinin eğitilmesinde kullanılan veri setleri hakkında bilgi verilmiştir. Bu veri setlerinin nereden temin edildiği ve örneklerle veri yapısı incelenmiştir. Üçüncü bölümde uygulamanın geliştirme aşamasındayken veriyi daha düzenli hale getirmek için kullanılan klasik doğal dil işleme metotları ve modellerin bazı önemli bölümleri hakkında sözde kod (pseudocode) ve akış diyagramı örnekleri ile birlikte bilgi verilmiştir. Dördüncü bölümde genel sistem tasarımı hakkında bilgi verilmiştir. Beşinci bölümde örnek metinler ile yapılan testlerin sonuçları incelenmiştir. Sonuncu bölümde ise uygulamanın gelecek araştırmalarda geliştirilebilecek özellikleri üzerinde durulmuştur.

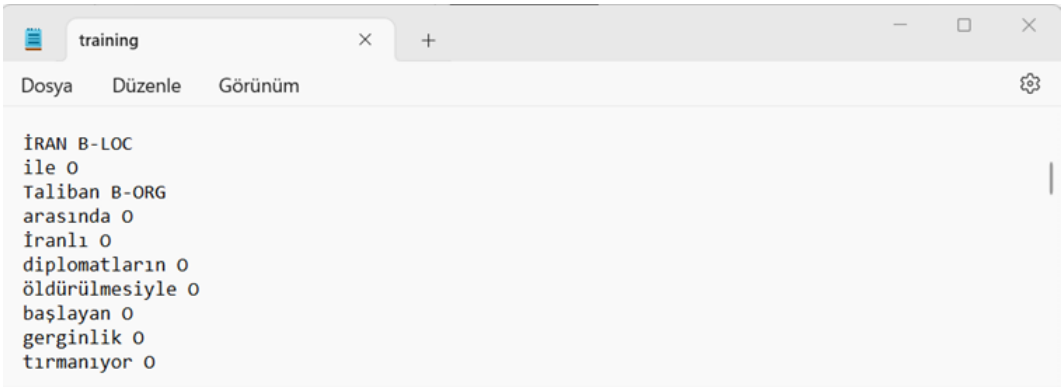
2
VERİ SETLERİ

Doğal dil işlemede yapay zeka modellerinin geliştirilmesi için kullanılan dönüştürücüler gözetimli öğrenme metodu ile eğitildiklerinden dolayı, bu modellerin test zamanı başarısı eğitimde kullanılan veri setinin kalitesine büyük ölçüde bağlıdır. Veri setinde olan dengesizlik veya veri kayıtlarının bir kısmının bozuk veya düzensiz olması modelin performansının düşmesine sebep olur. Oluşabilecek dengesizliğe misal olarak, veri setinde farklı sınıflara ait örneklerin sayıları arasında farkın büyük olduğu durumlar gösterilebilir. Böyle durumlarda eğitim zamanı alınan başarı puanının da önemi kalmaz. Bu sebepten eğitimden önce, kullanılacak veri setinin aykırı değer temizleme, veri artırma veya eksik veri doldurma gibi çeşitli yöntemlerle kalitesinin artırılması gerekir. Tez kapsamında geliştirilen projede iki derin öğrenme modeli: varlık ismi tanıma ve duygu analizi, bizim tarafımızdan eğitilmiştir. Bu modellerin eğitiminde yapısı farklı olan ve farklı kaynaklardan elde edilen iki büyük veri seti kullanılmıştır.

2.1 Varlık İsmi Tanıma Modeli için Kullanılan Veri Seti

Varlık ismi tanıma modelinin eğitimi için Serap Özkaya [14] tarafından oluşturulan bir veri seti kullanılmıştır. Bu veri seti her kelime ve etiket değeri tek bir satırda olup ve her cümle bir boş satır ile ayrılan .txt uzantılı bir dosyadan oluşmaktadır (Şekil 2.1).

Genel olarak, model eğitiminde “csv” formatındaki dosyaların kullanımı, “txt” formatındaki dosyaların kullanımından daha pratik olarak kabul edildiği için, “Python” dilinde yazılmış basit bit algoritma kullanılarak dosyanın formatı ve veri yapısı değiştirilmiştir. Bu değişimden sonra, veri seti “csv” dosyasının içinde, üç sütundan oluşmaktadır. Birinci sütun her satır için cümlenin indeksi, ikinci sütun cümledeki kelime, üçüncü sütun ise ikinci sütundaki kelimeye karşılık gelen, BIO formatında kodlanan etiketten oluşmaktadır. Veri seti toplamda 459.103 satır ve 26.378 cümleden oluşmaktadır (Şekil 2.2).



Şekil 2.1 Varlık ismi tanıma modelinin eğitimi için kullanılan veri setinin ilk versiyonuna ait görüntü

"labels" isimli sütunda yer bildiren 'B-LOCATION' ve 'I-LOCATION', tarih bildiren 'B-DATE' ve 'I-DATE', kişi özel ismi bildiren 'B-PERSON' ve 'I-PERSON', yüzde verilerini bildiren 'B-PERCENT' ve 'I-PERCENT', organizasyon ismini bildiren 'B-ORGANIZATION' ve 'I-ORGANIZATION', para markası bildiren 'B-MONEY' ve 'I-MONEY' ve zaman bildiren 'B-TIME' ve 'I-TIME' etiketleri yer almaktadır. Kelime cümlenin başında geldiği durumlarda etiketin önüne "B", ortada veya sonda geldiği durumlarda ise "I" harfi eklenmiştir. Özel bir kategoriye karşılık gelmeyen, diğer anlamına gelen kelimeler için "O" etiketi kullanılmıştır.

```
Data_train[Data_train['sentence_id']==28]
```

	sentence_id	words	labels
521	28	1994	B-DATE
522	28	genel	O
523	28	seçimlerinin	O
524	28	maliyeti	O
525	28	ise	O
526	28	98	B-MONEY
527	28	milyon	I-MONEY
528	28	672	I-MONEY
529	28	bin	I-MONEY
530	28	mark	I-MONEY
531	28	olarak	O
532	28	hesaplanmıştı	O

Şekil 2.2 Varlık ismi tanıma modelinin veri setinin değişimden sonraki versiyonuna ait görüntü

2.2 Duygu Analizi Modeli için Kullanılan Veri Seti

Çalışma kapsamında eğitilen duygu analizi derin öğrenme modeli için "VeriBilimiOkulu" [15] adresinden alınan veri seti kullanılmıştır. Veri seti "csv" dosya formatında indirilmiş, iki sütundan oluşmaktadır. İlk sütunda cümlenin kendisi, ikinci sütunda ise bu cümlenin duygusal tonunun eğilimini bildiren "0" veya "1" değeri bulunmaktadır (Şekil 2.3). "0" değeri negatif cümleleri, "1" değeri ise pozitif olanları bildirmektedir. Veri seti toplamda 31.514 cümleden oluşmaktadır. Bu cümlelerin 19.582 tanesi pozitif, 11.932 tanesi ise negatif duyguya eğilimli cümlelerdir.

	text	category	encoded_categories
29186	heyecanla aldık.. Soğan ile başladık.. ama ne ...	0	0
23969	Ürünü ggıdıyordan aldım 1 kez bile kullanamada...	0	0
5490	banka kartımı postacı değilde kendi geliyor sanki	0	0
6229	Telefon harika kamerası bataryası çok iyi. Gün...	1	1
26243	her yönüyle çok iyi bir ürün yalnız ben 240 a ...	1	1
9959	Daha önceleri i başka marka bluetooth kulaklık...	1	1

Şekil 2.3 Duygu analizi modelinin veri setinin bir bölümüne ait görüntü

Görseldeki "encoded categories" sütununda, "category" sütunundaki "metin" tipli verilerin "tam sayı" tipinde olan versiyonları bulunmaktadır. Veri tipleri arasındaki bu geçiş "sklearn" kütüphanesinin "LabelEncoder (). fit transform" metodu ile sağlanmıştır.

2.3 Veri İşleme

Model eğitiminden önce, hem varlık ismi tanıma hem de duygu analizi modellerinin veri setleri için, veri temizleme işlemi yapılmıştır. İlk olarak sütunlardan biri için boş değer taşıyan satırlar, eğitimin performansına negatif yönde etki etmemesi için belirlenip, veri setinden silinmiştir. Bu aşamadan sonra veri setlerinin her satırı kontrol edilip, gereksiz olan boşluklar silinmiştir. Son olarak, nokta istisna olmak üzere diğer özel karakter ve işaretler cümlelerden kaldırılmıştır.

3
YÖNTEM

Bu bölümde, çalışmada varlık ismi tanıma ve duygu analizi aşamalarında kullanılan programlama dili, kütüphanelerin ve derin öğrenme modellerinin özellikleri detaylı bir şekilde açıklanmıştır.

3.1 Python

Geliştirilen uygulamada programlama dili olarak Python kullanılmıştır. Genel olarak Python yapay zeka projelerinde yaygın olarak tercih edilmektedir. Buna sebep olarak Pythonun diğer programlama dillerine nazaran basit ve esnek olması gösterilmektedir. Bu programlama dilinde veri yapılarının dinamik olması ve sentaksının öğrenilmesinin rahat olması, veri bilimciler tarafından seçilmesini sağlamaktadır. Yapay zeka projelerinde verinin kendisi ve model eğitimi esas odak noktası olduğundan, programlama dilinin nispeten daha rahat okunabilir olması yazılımcılar için kolaylık sağlamaktadır. Özellikle verilerin eğitim aşaması için uygun formata dönüştürülmesi bu programlama dilinde çok hızlı ve basit şekilde yapılabilmektedir. Bir diğer sebep ise Python'un veri bilimi için çok zengin kütüphanelere sahip olmasıdır. Misal olarak, verilerin dizi veya tensör olarak ifade edilmesinde Pythorch veya Tensorflow kütüphaneleri kullanılmaktadır. Derin öğrenme ve klasik makine öğrenmesi modelleri eğitim aşamasında girdi veri yapısı olarak tensörleri kabul ederler. Özellikle bu kütüphanelerin içinde yapay zeka modellerinin eğitilmesi için de metodlar bulunmaktadır. Bundan başka Pythonun spesifik olarak “tokenization”, kelimelerin köklerinin alınması veya bağlılık ayrıştırması gibi doğal dil işleme projeleri için kullanılan kütüphaneleri de mecvuttur. Python programlama dilinin kullanıcı kitlesinin çok geniş olması bu kütüphanelerin sürekli olarak güncellenmesini ve geliştirilmesini sağlamaktadır.

3.2 Pytorch

Pytorch [16] kütüphanesi yapay zeka projelerinde verilerin tensörlere dönüştürülmesi, hem bu tensörler üzerinde kompleks matematiksel işlemlerin yapılması, hem de kullanılan makine öğrenmesi veya yapay zeka modelinin parametrelerinin düzenlenmesinde kullanılır. Tensörler çok boyutlu diziler halinde

projede kullanılan veriyi temsil ederler. Örnek olarak, siyah beyaz bir resimin piksellerini 2 boyutlu tensörlerle ifade etmek mümkündür. Renkli görüntüde ise boyut 3 olarak ayarlanır. Bu kütüphanenin sağladığı bazı metodlar bu tipli kompleks veriler üzerinde çarpma, kare ve b. gibi işlemleri kolaylaştırmaktadır.

Doğal dil işleme projelerinde ise bu tensörler, metinleri veya cümleleri sayısal veriler şeklinde diziler halinde tutarlar. Çalışma kapsamında, duygu analizi aşamasında modelin eğitimi için kullanılan veri setinde cümleler token'lara ayrıldıktan sonra bu token'lar sayılara kodlanır ve Pytorch ile bir tensörün içine toplanır. Bu aşamada artık eğitim verisi BERT [17] dönüştürücü modelinin kabul ettiği formata çevrilmiştir. BERT modelinin "Huggingface" platformundan sınıflandırma amaçlı yüklenmesi için de Pytorch kullanılmıştır.

Bu kütüphanenin bir diğer kullanım alanı da modelleri eğittikten sonra bilgisayara kaydedip sonradan yüklemektir. Pytorchun bu özelliği sayesinde model tüm eğitim parametreleriyle birlikte yaklaşık 500 – 900 megabayt arası ölçüde ".model" uzantılı bir dosyanın içinde bilgisayarın hafızasına kaydedilmektedir. Sonradan bu model, yine bu kütüphane ile başka bir proje dosyasının içeriğine yüklenip kullanılabilir.

Derin öğrenme modellerinin eğitilmesinde GPU ile eğitme seçeneğinin varlığı önem arz etmektedir. Eğitim aşamasında aktivasyon fonksiyonları hesaplamalarının veya büyük boyutlu matris çarpımlarının GPU'ların paralel hesaplama özelliği ile yapılması eğitim süresini önemli ölçüde kısaltabilmektedir. Çalışmada Pytorch ile eğitim için GPU – cuda seçilmiştir.

3.3 Zeyrek (Zemberek)

Zemberek doğal dil işleme projelerinde, Türkçe metinler üzerinde kelime kökünün eklerin silinerek bulunması (stemming), kelimenin kökünün anlamsal olarak tespit edilip bulunması (lemmatization), cümle ayrıştırma, metindeki yazım hatalarını tespit etme gibi işlemlerin uygulanmasında kullanılan kütüphanedir. Bu tür işlemler özellikle varlık ismi tespiti, metin özetleme, duygu analizi gibi konularda çok sıklıkla kullanılmaktadır ve böyle bir kütüphanenin Türkçe için bulunması kolaylık sağlamaktadır. Etkili bir doğal dil işleme modeli eğitmek için metinlerdeki kelimelerin yapılarının veya formlarının analiz edilmesi önemli bir aşamadır. Misal

olarak, metindeki yazım hatalarının bulunmasında kelime köklerinin ve eklerinin analiz edilmesi gerekmektedir. İlaveten, metin sınıflandırma gibi bilgi çıkarma uygulamalarında metinlerin hangi kategoriye ait oldukarının tespit edilmesi için o metindeki kelimelerin köklerinin incelemesi, uygulamanın odak noktalarından biridir. Çalışmada Zemberek kütüphanesinin "lemmatizing" ve "stemming" için kullanılan ve Python dilinde çalışan "Zeyrek" [18] aracı kullanılmıştır.

Stemming kelimenin kökünün sonundaki eklerin silinerek bulunma yöntemidir. Misal olarak "Arabalar" sözünden "-lar" ekini silerek "Araba"-yı, "Şehirlerin" sözünden ise "-lerin" ekini silerek "Şehir"-i buluruz. Ancak bazı sözler için stemming işlemi başarılı sonuç veremiyor. Bu durumlar için örnekler esasen ingilizcede bol olsa da, türkçede de bazı kelimeler için göstermek mümkündür. Misal olarak, "kitabın" kelimesinin kökünü stemming işlemi ile bulmaya çalışırsak bize sonda çıktı olarak "kitab" verecektir. Çünkü stemming işleminin çalışma prensibi kelime üzerinde algoritmik olarak ve dilbilgisi kullanarak ekleri belirleyip onları silmektir.

Lemmatizing aynı şekilde kelimenin kökünün bulunması için kullanılan fakat stemming yönteminin yetersiz kaldığı örnekler için çalışan bir metoddur. Bu yöntem stemming işleminden farklı olarak sözlük bilgisi veya kelime ağı da kullandığı için daha başarılı çalışmaktadır. Kelime ağları kelimelerin eş ve karşıt anlamlarını, diğer formlarıyla olan ilişkilerini gösteren özel bir yapısı olan sözlük gibi düşünülebilir. Eğer kelimenin kökünün bulunması işlemi basit dil bilgisi kuralları ile sağlanamazsa lemmatizing yöntemi kelime ağlarını kullanır. Doğal dil işleme projelerinde sıkça kullanılan kelime ağına misal olarak "WordNet" gösterilebilir. "WordNet" ağı kelimelerin eşanlam veya karşıt anlam gibi bilgilerini kendinde tutmaktadır. Bu kelime ağı Princeton Üniversitesinde geliştirilmiş olup, doğal dil işleme projelerinde sıkça kullanılmaktadır. Tez projesinde kelimelerin köklerinin bulunması için lemmatizing işlemi kullanılmıştır.

3.4 Spacy

Spacy [19] doğal dil işleme projeleri için tasarlanmış en kapsamlı Python kütüphanelerinden biridir. Kelimelerin morfolojik analizi – isimlerin veya fiillerin bulunması, cümlelerin token'lara ayrılması, lemmatizing işlemi ile kelime köklerinin bulunması gibi temel işlemler bu kütüphane ile yapılabilmektedir.

Bu kütüphane, kelimeleri doğal dil işleme modellerinin girdi olarak kabul ettiği vektör formatına dönüştürmek için kelime gömme metodlarını da içermektedir. Bu vektörler aynı zamanda kelimenin anlamı ve başka kelimelerle olan benzerlik ilişkilerini de tutmaktadır. Spacy önceden eğitilmiş olan modeller ile veri setindeki kelimeler için bu vektörleri çıkarabilmektedir.

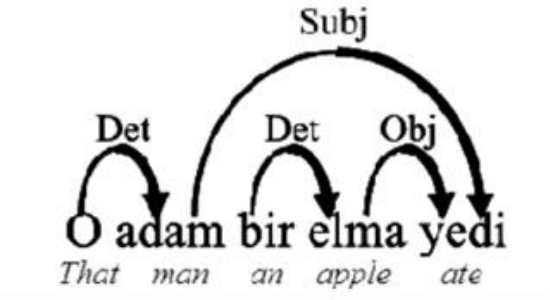
Spacy kütüphanesi metin sınıflandırma, duygu analizi veya varlık ismi tanıma gibi konular için de kullanılabilmektedir. Bu işlemleri gerçekleştirmek için kütüphane önceden kaliteli ve etiketli veri ile eğitilmiş olan yapay zeka modellerini kendinde saklar. Uygulama esnasında derin öğrenme modelinin kaydedilmiş ağırlıklarını kullanarak görevi gerçekleştirir.

```
# turkce icin : "tr_core_news_sm"
nlp = spacy.yukle("tr_core_news_sm")
# Metin
text = "Istanbul cok guzel bir sehir."
# Metni isleyin
doc = nlp(text)
```

Şekil 3.1 Spacy kütüphanesinin yüklenmesine ait sözde kod

Şekil 3.1'deki "doc" değişkeni örnek cümlenin token veya öğe bilgilerini tutmaktadır. Bu değişken yerine göre hem varlık ismi tanıma gibi kompleks bir görev için hem de cümleyi tokenlere ayırmak için kullanılabilir.

Bu çalışmada Spacy kütüphanesi uygulamanın kilit noktalarından biri olan bağlılık ayrıştırması görevi için kullanılmıştır (Şekil 3.2). Kütüphane bu işlemi yukarıda bahsedilen duygu analizi veya varlık ismi tanıma görevine benzer olarak kendisinde bulunan ve önceden eğitilmiş modeli kullanarak gerçekleştirir. Çalışmada masallardaki varlık ismi tanıma modelinin bulmakta başarısız olduğu özneleri tespit etmek için kullanılmıştır.



Şekil 3.2 Bağlılık ayrıştırma yöntemine ait bir örnek

Şekil 3.3'de ".dep" döngüde sırası gelen kelimenin cümle içerisindeki bağlılık ilişkisini göstermektedir. Bu bağlılık ilişkisi kelime ile cümledeki diğer kelimeler arasında olan çokyönlü bir ilişkidir. 'nsubj' kelimenin özne olduğunu, 'ROOT' ise kelimenin kök formatında olduğunu gösteren bağımlılık ilişkisi anahtar sözcüğüdür.

```
dongu icinde doc:
        eger (('nsubj' in sent.dep_) or ('ROOT' in
           sent.dep_))
```

Şekil 3.3 Çalışmada kullanılan bağlılık ayrıştırmasına ait sözde kod

3.5 Metin Temsili Yöntemleri

Doğal dil işleme projelerinde kilit adımlardan biri eğitim için kullanılan metin verilerinin sayısal formata çevrilerek uygulanacak olan derin öğrenme modelinin kabul ettiği yapıya getirilmesidir. Bu adım için bir kaç temel teknik bulunmaktadır. Bu teknikler bir birinden bazı farklı ve önemli özellikleri taşıması ve basitlik derecesine göre farklılaşmaktadır. Bu teknikler ayrık ve sürekli metin temsilleri olarak iki türe ayrılmaktadır. Ayrık metin temsili yöntemleri doküman içerisindeki kelimelerin sıklıklarını hesaplayarak vektör yapısı oluşturur.

3.5.1 Bag of Words

İlk ayrık yöntem olan" Bag of Words" (BoW) modeli basit bir şekilde metin içerisindeki bulunan her kelime için bir sütun, bu metindeki her cümle için satırdan oluşan bir matris oluşturur. Bu yöntemde kelimelerin sırası tutulmaz ve kelime sıklıkları ile metinler arasındaki benzerlikleri tespit etmek için kullanılır. Basit ve anlaşılabilir olması kolaylık sağlasa da bu yöntemde kelimeler sırasız ve bağımsız

bir şekilde temsil edilir (Şekil 3.4). Bu sebepten "BoW" modeli ile kelimelerin birbiriyle anlamsal bağını ifade etmek mümkün değildir.

```
dongu icinde doc:
        eger (('nsubj' in sent.dep_) or ('ROOT' in
           sent.dep_))
```

Şekil 3.4 BoW" yöntemi için örnek

3.5.2 TF-IDF

Belli bir kelimenin belli bir metin dosyası ile ne kadar ilgili olduğunu ifade etmenin bir diğer yolu da "TF-IDF (Term Frequency-Inverse Document Frequency)" dir. Bu metodda kullanılan bazı formüller sayesinde kelimelerin metin içerisindeki önemi hesaplanabilmektedir.

İki aşamadan oluşan yöntemde ilk olarak kelimenin cümle veya belgedeki (formülleri izah etmek için resmi terim olarak genellikle belge kullanılmaktadır) frekansı hesaplanmaktadır (Eşitlik 3.1).

$$\mathrm{TF}(t,d) = \frac{\text{"d" } belgesindeki \text{ "t" } teriminin\ geçiş\ sayısı}{\text{"d" } belgesindeki\ toplam\ terim\ sayısı} \tag{3.1}$$

Bu aşamadan sonra ise kelimenin tüm belgelerdeki kullanma sıklığı hesaplanıp, ne kadar bilgi taşıdığı tespit edilir. Kelime nadir olarak kullanılmışsa daha çok bilgi taşımaktadır. İkinci aşamanın kullanılmasının esas odak noktası "Stop words" gibi anlam taşımayan kelimelere yüksek değer atanmasının önlenmesidir (Eşitlik 3.2).

$$\mathrm{IDF}(t,D) = \log\left(\frac{\text{Toplam belge sayısı (D)}}{\text{"t" terimini içeren belge sayısı}}\right) \tag{3.2}$$

Son adımda ise TF ve IDF değerlerinin çarpımından kelimenin metin içerisindeki değeri bulunur (Eşitlik 3.3).

$$\text{TF-IDF}(t,d,D) = TF(t,d) \times IDF(t,D) \tag{3.3}$$

3.5.3 One Hot Encoding

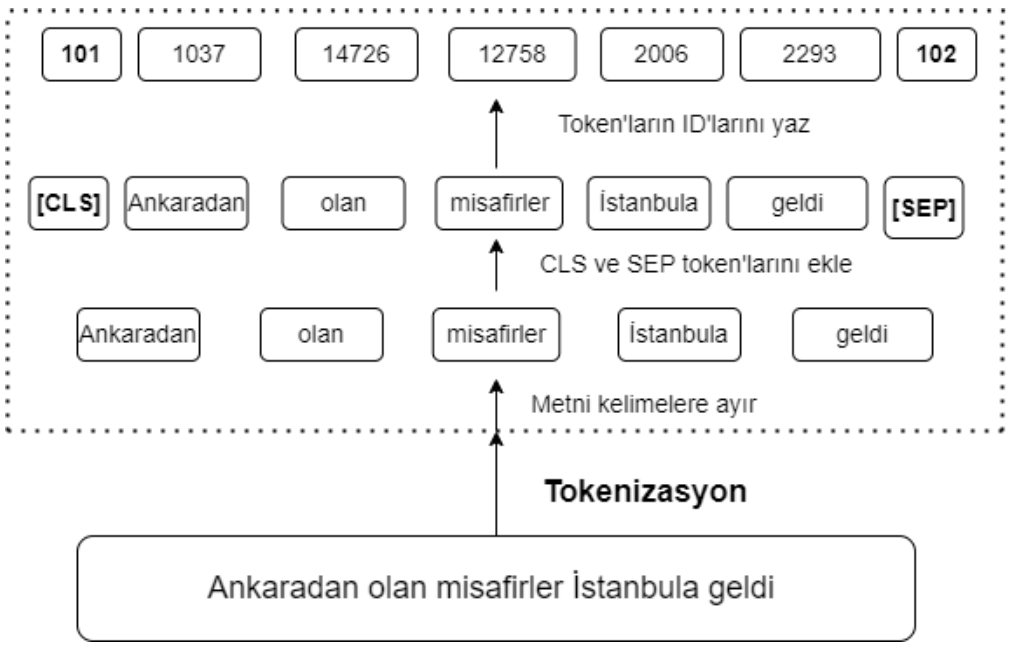
Veri setindeki kategorik verileri vektörlerle ifade etmenin yollarından biri "One Hot Encoding" metodudur. Bu metodda vektörün uzunluğu farklı kategorik verilerin

sayısına eşittir. Misal olarak, veri setinde 4 farklı renk varsa, bu durumda vektörün uzunluğu da 4 olacaktır. Vektörün içinde spesifik kategoriyi temsil eden indeks 1, geri kalanları ise 0 ile temsil edilir.

BoW modeli gibi anlaşılabilir olsada, bu metodun dezavantajı büyük hacimli kategori verisine sahip setler için vektörlerin boyutunun büyümesi hesaplamalarda maliyetin artmasına neden olur.

3.6 Tokenizasyon

Doğal dil işleme modellerinin eğitiminde esas adımlardan biri de tokenizasyon işlemidir. Cümleyi token'lara ayırma işlemi de kelime gömme gibi BERT dönüştürücüsünün kendi içinde yapılabilmektedir. Tokenizasyon işleminin sonucunda cümle kelime ve sayılarla ifade edilen ve modelin bir sonraki aşamalar için kullanacağı token'lara dönüştürülür. Sıra ile tüm kelimeler token'lara dönüştürüldükten sonra cümlenin başına "CLS" – 101, sonuna ise "SEP"-102 token'ları eklenir. Bu token'ların görevi modele cümlenin nerede başlayıp nerede bittiği bilgisini vermektir (Şekil 3.5).

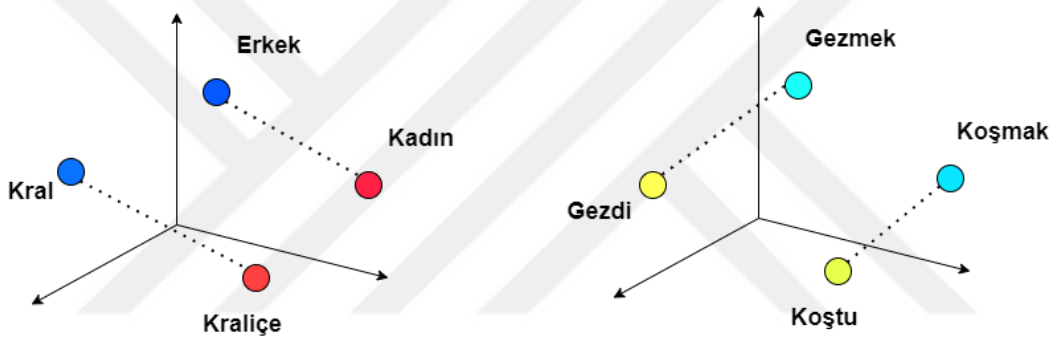


Şekil 3.5 BERT tokenizasyon aşamasına ait örnek bir görüntü

3.7 Kelime Gömmeleri

Kelime gömmeleri metindeki kelimelerden çokboyutlu kelime uzayı yaratmak için kullanılan ileri seviye bir yöntemdir. Bu metodda belgedeki her kelime bir vektörle ifade edilmektedir. Bu vektörler kelime uzayında her kelimenin semantik anlamını ve diğer kelimelerle olan ilişkilerinin bilgilerini taşımaktadır. Yaratılan kelime uzayında anlamsal olarak benzer olan kelimeler birbirine yakın, farklı olanlar ise uzakta konumlandırılırlar. Misal olarak "erkek" ve "kadın" sözleri birbirine yakın konumda olurken, "erkek" ve "ülke" kelimeleri birbirinden uzakta yerleşirler.

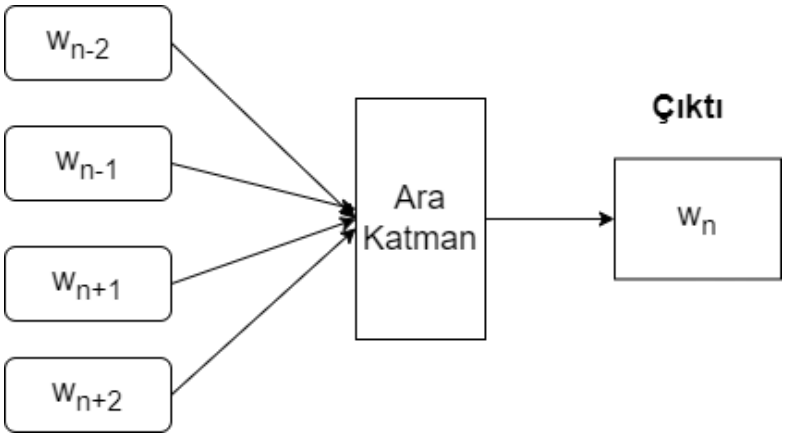
Kelime gömmeleri derin öğrenme yöntemi ile geliştirilmiş olan yapay sinir ağları ile uygulanır. Bu modellerden sıklıkla kullanılanlara örnek olarak "Word2Vec" veya "Glove (Global Vectors for Word Representation)" gösterilebilir (Şekil 3.6).



Şekil 3.6 Kelime gömme ile yaratılan çokboyutlu kelime uzayı ait görüntü

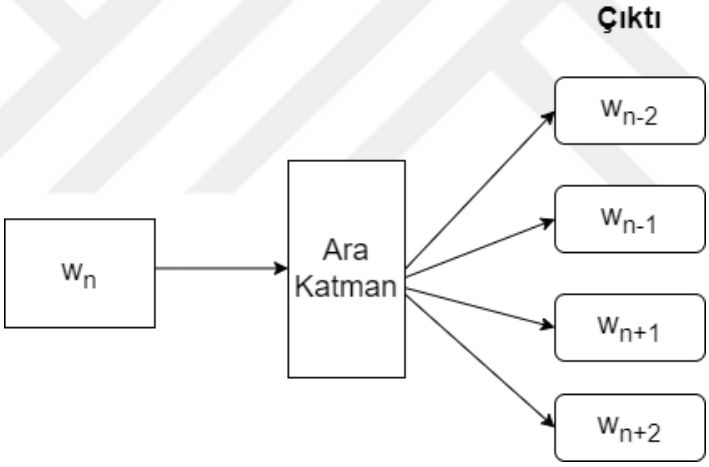
Word2Vec [20] özünde bir yapay zeka sinir ağı modelidir. Bu model büyük bir veri seti ile eğitildikten sonra modelin içinde bulunan katmanlardaki ağırlıklarda kelimelerin vektör temsilleri tutulur. Word2Vec modeli metin sınıflandırma gibi bir çok doğal dil işleme uygulamalarında kelimeleri onların semantik anlamlarıyla birlikte vektörel olarak ifade etmek için kullanılmaktadır. Word2Vec modelinin 2 ana metodu bulunmaktadır: "Skip-gram", "CBOW".

İlk yöntem olan CBOW (Continuous Bag of Words) bilinmeyen kelimeyi tahmin etmek için cümle içinde bulunan diğer kelimelerin vektörel bilgilerini kullanır. Kullanılan diğer kelimeler ise cümle içinde aranan kelimenin etrafında yerleşen kelimelerdir (Şekil 3.7).



Şekil 3.7 CBOW yöntemine ait basit bir diagram

İkinci metot olan "Skip-gram" ise verilen kelimenin bilgilerini kullanarak etrafındaki kelimeleri tahmin etmek için kullanılan algoritmadır. Skip-gram metodunda CBOW yönteminde olduğu gibi cümledeki her kelime için sıra ile vektör değeri hesaplanır (Şekil 3.8).

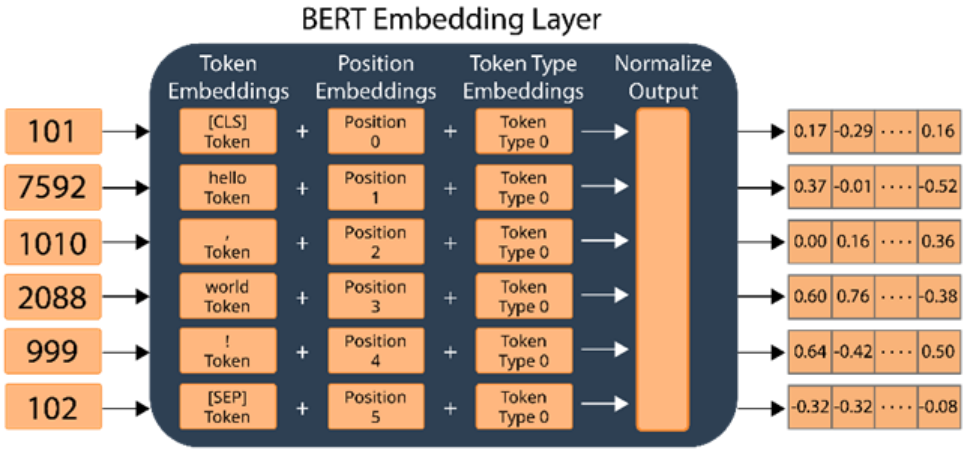


Şekil 3.8 Skip-gram modelina ait basit bir görüntü

Çalışmada kelime gömme işlemi, kullanılan BERT dönüştürücüsünün gömme katmanında yapılmaktadır. İlk olarak, cümle dönüştürücü tarafından tokenlere ayrılır. Sonraki adımda bu tokenlerin kendi bilgileri, cümledeki pozisyonlarının bilgileri ve metin içerisindeki segment ayrımı bilgilerini kapsayan bir vektörel değer üretilir. Bu değer kelimenin kendi anlamını, hem cümledeki diğer kelimelerle olan bağlılıklarını hem de cümlenin kendi yapısını temsil eder.

Bu metod "Word2Vec" yöntemine nazaran biraz daha karışık olsa da vektörler yine aynı şekilde cümledeki maskelenen, yani bilinmeyen kelimeyi tahmin ederek bulunur. BERT'in üstün tarafı ise metinde birden fazla segment üzerinde aynı anda

çalışabilmesidir. Bu özellik metindeki cümleler arasındaki ilişkiyi anlama açısından önemlidir (Şekil 3.9).



Şekil 3.9 BERT gömme katmanına ait bir görüntü [21]

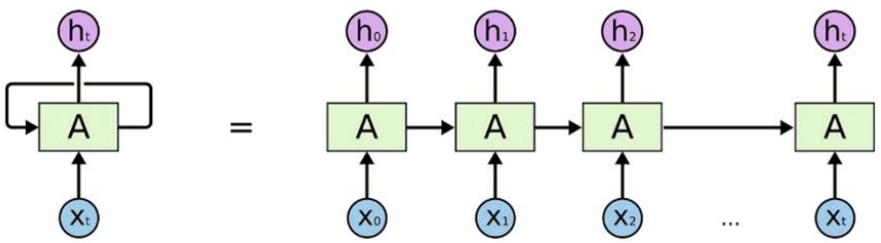
3.8 Derin Öğrenme Teknikleri

Günümüzde, doğal dil işleme çalışmalarında derin öğrenme modellerinden tekrarlayan sinir ağları (RNN), LSTM ve daha verimli olan, dikkat mekanizması ile çalışan dönüştürücüler kullanılmaktadır.

3.8.1 RNN (Tekrarlayan sinir ağları), LSTM

Tekrarlayan sinir ağları [22] normal yapay sinir ağlarından farklı olarak zaman bağlılığı veya sıra mantığı içeren konularda tahmin uygulamalarının geliştirilmesi için kullanılır. RNN kendisinde bulunan hafıza sistemi sayesinde eğitim aşamasında geçmişteki adımlarda elde edilen bilgileri hatırlayabilmektedir. Bu bilgileri mevcut döngüdeki çıktıyı belirlemek için kullanır. Yapay sinir ağı bu işlemi, ara katmanlarda hücre durumunu (h_t) güncellemek için önceki hücre durumu (h_{t-1}) ve girdi verisi (x_t) bilgilerini kullanarak yapmaktadır (Eşitlik 3.4).

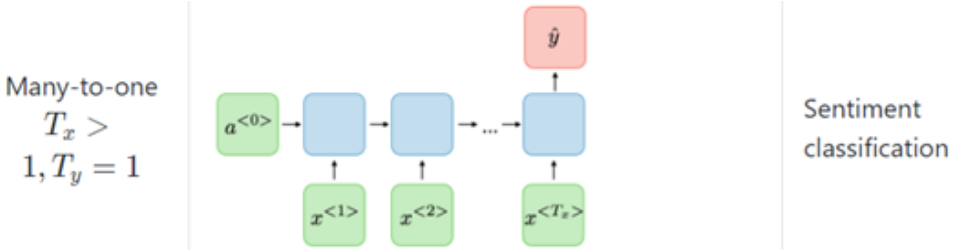
Bu sebepten RNN yapısında girdi verileri aslında birbiriyle ilişkilidir ve bu ilişkiyi kurabilmek için döngüsel bir yapı kullanılır (Şekil 3.10).



Şekil 3.10 Tekrarlayan sinir ağının yapısına ait basit bir görsel [23]

$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t) \tag{3.4}$$

Doğal dil işleme çalışmalarında kullanılan metin verileri veya cümleler tokenizasyon işleminden geçtikden sonra dizi olarak temsil edilirler. RNN'ler bu diziler üzerinde çalışarak cümledeki kelimeler arasındaki ilişkileri öğrenir. Kelimelerin cümledeki anlamları onların birlikte kullanıldığı diğer kelimelere bağlıdır. Cümlenin kendi anlamı veya duygusal tonu bu cümlenin başında veya sonunda kullanılan bir kelime tarafından değişebilir. Bu sebepten, duygu analizi, varlık ismi tespiti, bağlılık ayrıştırması gibi görevlerde yapay sinir ağının cümledeki ilişkileri öğrenmesi önem arz etmektedir. Zaman kavramı içeren konularda olduğu gibi doğal dil işleme projelerinde de RNN'ler hafıza özelliğini kullanarak cümledeki bağımlılıkları çözümleyebilmektedir (Şekil 3.11).

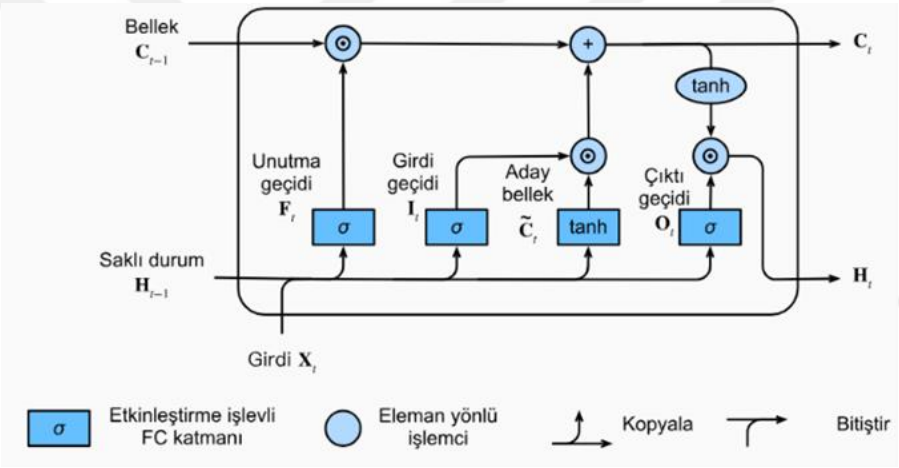


Şekil 3.11 RNN yapısına ait basit bir görsel [24]

RNN'ler doğal dil işlemede önemli konuma sahip olsa da, geriye doğru yayılım aşamasında (backpropagation) gerçekleşen kaybolan ve patlayan gradyan sorunu bu modellerin uzun cümlelerdeki bağımlılıkları öğrenmesini önemli derecede kısıtlar ve eğitim kalitesini düşürür. Bu sorunun çözümü için LSTM ve GRU olarak isimlendirilen gelişmiş tekrarlayan sinir ağı modellerinin kullanımı tercih edilmektedir.

LSTM (Uzun Kısa Süreli Bellek) sinir ağı modelinde RNN'den farklı olarak giriş, çıkış unutma diye isimlendirilen kapılar ve hücre mevcuttur (Şekil 3.12). Hücre

bileşeni sinir ağı modelinde zaman adımlarında güncellenir. Bu bileşen sinir ağının döngülerde elde etdiği bilgileri taşır ve LSTM'in uzun süreli hafızasını temsil etmektedir. Modeldeki unutma kapısı her döngüde hücrede bulunan bilgilerin ne kadarının korunacağını ve ne kadarının unutulacağını belirlemek için bulunur. Bu kapı hücreden saklanmaması gereken bilgileri çıkarır. Diğer bileşen olan giriş kapısı ise hücreye hangi bilginin gireceğini belirlemek için mevcuttur. Yeni eklenen bilgilerle birlikte hücre güncellenir. Son olarak, çıktı kapısı hücredeki bilgilerin modelin çıktısına eklenmesini kontrol eder. Bu üç bileşenin her zaman adımında veya döngüde sinir ağı modelindeki bilgiler için kullandıkları kontrol mekanizması sayesinde LSTM modeli kaybolan veya gradyan probleminin üstesinden gelebilmektedir. Bunun sayesinde, LSTM uzun vadeli bağımlılıkların öğrenilmesinde daha iyi performans gösterebilmektedir.



Şekil 3.12 LSTM ağının giriş, unutma ve çıktı geçitlerine ait görsel [25]

Bu kapıların *(t)* anındaki değerlerinin hesaplanmasında sigmoid fonksyonu kullanılır. Eşitlik 3.5'de giriş kapısının *(t)* anındaki değerinin hesaplanması gösterilmiştir. Burada *(W)* ağın ağırlık değerlerini , *(H)* önceki döngülerin gizli durumunu, *(X)* girdi verilerini, *(b)* ise girdi parametresini ifade etmektedir.

$$I_t = \sigma(W_{xi}X_t + W_{hi}X_{t-1} + b_i) \tag{3.5}$$

3.8.2 Dönüştürücüler

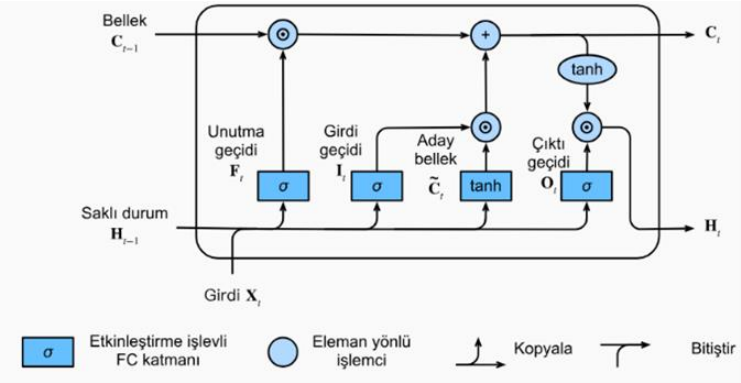
LSTM modeli RNN'de bulunan gradyan sorunlarına çözüm olarak uygulansa da bir kaç aşamadan oluşan kontrol mekanizmasından dolayı hızı RNN'den bile yavaştır. Bu durum eğitim aşamasında uzun cümlelerin veya metinlerin öğrenilmesi zamanı dezavantaj olarak değerlendirilmektedir. İlk defa 2017 yılında “Attention is All You

Need" isimli popüler makale ile tanıtılan dönüştürücüler doğal dil işleme projelerinde yapay zekanın uygulanmasına olan bakış açısını büyük ölçüde değiştirmiştir. Buna sebep ise dönüştürücülerin RNN veya LSTM'de bulunan sıralı hesaplama sistemini kullanmak yerine daha ileri seviye bir metod olan dikkat mekanizmasını kullanmasıdır. Bu mekanizma ve paralel hesaplama özelliği sayesinde dönüştürücüler doğal dil işleme alanında çök büyük bir öneme sahiptir.

Dikkat mekanizması, dönüştürücü bir görevi yaparken girdi olarak verilen metnin veya cümlenin hangi bölümlerinin daha fazla öneme sahip olduğunu belirlemektedir. Bu yöntem sayesinde yapay zeka modeli metinler üzerinde odaklı çalışarak daha verimli sonuçlar üretebilmektedir. Sıralı verilerde – özellikle cümlelerde – öğeler arasında her zaman belli bir bağlantı vardır ve bu mekanizma matematiksel işlemlerle cümle içerisindeki kelimelerin birbiriyle olan anlamsal bağını da hesaplayabilmektedir. Özellikle, özdikkat (self-attention) cümledeki bileşenler arasındaki ilişkiyi değerlendirmek için kullanılır. Özdikkat bu işlemi cümle içindeki her söz için bir ağırlık değeri hesaplayarak gerçekleştirir. Bu ağırlık değerleri, sonraki aşamada hangi kelimenin daha fazla anlam taşıdığını belirtmek için kullanılır. Böylece yapay zeka modeli cümlenin anlamını daha iyi seviyyede öğrenebilmektedir.

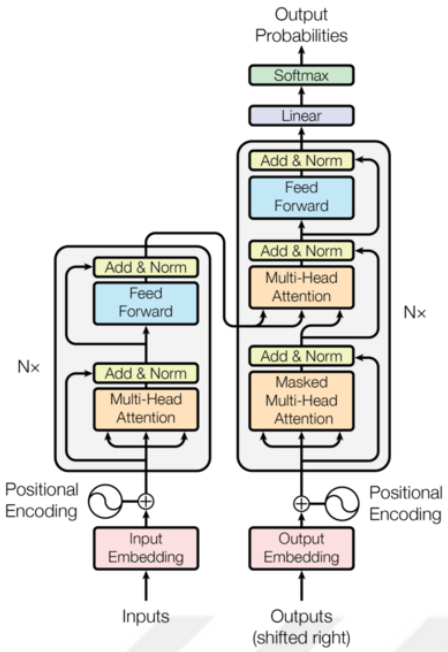
$$Attention(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V \tag{3.6}$$

Eşitlik 3.6'de gösterilen *(Q), (K), (V)* matrisleri cümledeki her kelime için ayrı-ayrılıkta oluşturulmaktadır ve bu matrisler sırasıyla sorgu, anahtar ve değer matrisi olarak isimlendirilir. Dikkat mekanizması ağırlık toplama işleminde sorgu ve anahtar vektörlerinin benzerliklerini ölçerek hangi kelimenin anlam değerinin daha fazla olduğunu belirler. Buradaki *(Q, (K), (V)* matrisleri "input embedding" aşamasında ve "hidden state" olarak isimlendirilen birimde türetilir. Formüldeki kök altındaki (d_k) ise anahtar vektörünün boyutunu temsil etmektedir. Bu formül son adımda "softmax" fonkstonu ile dikkat ağırlıkları arasındaki ilişkiyi gösteren ve [0,1] arasında olan bir değer üretir. Dönüştürücülerde bulunan çoklu-kafalı dikkat mekanizması daha verimli sonuç elde etmek için kullanılır. Bu mekanizma bir kelimenin önemini hesaplarken birden fazla özdikkat mekanizmasını kullanarak çıktıları birleştirip sonuç üretir (Şekil 3.13).



Şekil 3.13 Çok-kafalı dikkat mekanizmasına ait basit bir görsel [26]

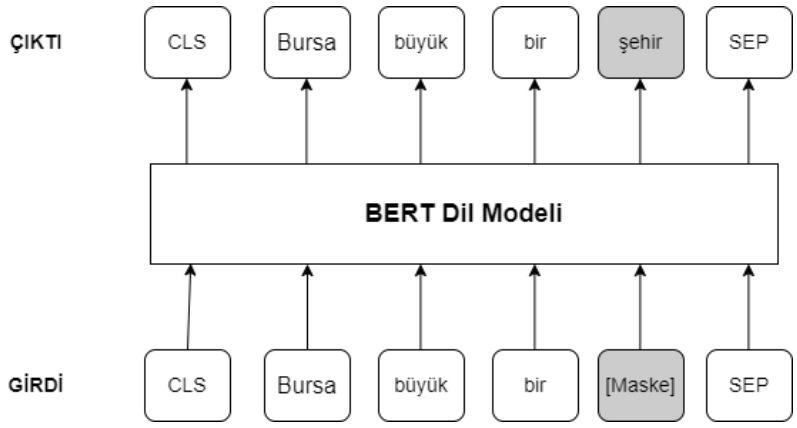
Dikkat mekanizmasından başka, dönüştürücü mimarisinde bulunan pozisyon kodlaması aşaması girdi verisindeki her kelimenin cümle veya metin içerisindeki pozisyon bilgisini modele vermek için kullanılır. Diğer adım olan ekleyip normalleştirme (add norm) her katmandan elde edilen çıktıya bir önceki katmanın çıktısını ilave edip normalleştirme aşamasını uygulamak içindir. Bu adım aşırı öğrenme veya ezberleme riskini azaltdığı için dönüştürücü mimarisinin önemli bileşenlerinden biridir. Genel olarak, dönüştürücüler kodlayıcı (encoder) ve kod çözücü (decoder) olarak isimlendirilen iki bölümden oluşmaktadır (Şekil 3.14). Kodlayıcı bölümü girdi verisini ifade etmek için vektörler üretir. Bu vektörler cümlenin veya metnin anlamını kendinde saklamak içindir. Diğer taraftan, kod çözücüler ise kodlayıcıdan gelen verileri inceleyip yeni veri dizisi üretmek için kullanılır. Özellikle, metin oluşturma işlemini kapsayan konularda kod çözücü bileşeni büyük önem taşımaktadır. Metin veya cümle üretilmesi gerekmeyen görevler için sadece kodlayıcı kullanılarak dönüştürücü programlanabilir ve bu durumda kodlayıcıdan gelen çıktılar doğrudan kullanılır. Bu görevlere misal olarak, duygu analizi veya varlık ismi tanıma gösterilebilir.



Şekil 3.14 Dönüştürücü mimarisine ait görsel [13]

Çalışmada varlık ismi tanıma ve duygu analizi modellerinin geliştirilmesi içim dönüştürücü olarak BERT kullanılmıştır. Derin öğrenme modeli olarak son derece etkili olan BERT dönüştürücüsü büyük hacimde bir veri setiyle eğitilir. Eğitim aşamasında metinde olan kelimeler arasındaki ilişkiler BERT tarafından belirlendikten sonra eğitim verisinin yapısına göre model farklı görevler için kullanılabilir. BERT modeli genel dönüştürücü yapısının sadece kodlayıcı (encoder) kısmını kullanmaktadır. BERT varlık ismi tanıma, duygu analizi, metin sınıflandırma, çeviri gibi konularda oldukça yüksek verimlilikte çalışmaktadır. Modelin başarısı eğitim için kullanılan veri setine bağlı olsa da, BERT modelinin bazı kendine özgü, avantaj kazandıran özellikleri bulunmaktadır. Kelimelerin cümle içindeki bağlamlarının tespit edilmesi zamanı iki yönlü dikkat mekanizması kullanılması bu modelin yüksek verimlilikte çalışması nedenlerinden bir tanesidir. İki yönlü dikkat mekanizması kelime için dikkat ağırlıklarının hesaplanması aşamasında cümle içinde hem önceki hem de sonraki kelimelerin dikkate alınmasıdır. Bunun sayesinde model kelimeler için daha kapsamlı anlam çıkarımı yapabilmektedir. BERT dönüştürücü modelinin ikinci özelliği ise eğitim aşamasında maskeleme tekniğini kullanmasıdır (Şekil 3.15). Eğitim aşamasında her adımda rastgele bir söz maskelenir ve model bu maskelenen sözü tahmin etmek için eğitilir. Bu özellik dönüştürücünün metin içindeki kelimeler arasındaki bağlamları

daha iyi öğrenmesine ve eğitim aşamasında karşılaşabileceği belirsizliklerle başa çıkmasına yardımcı olur.

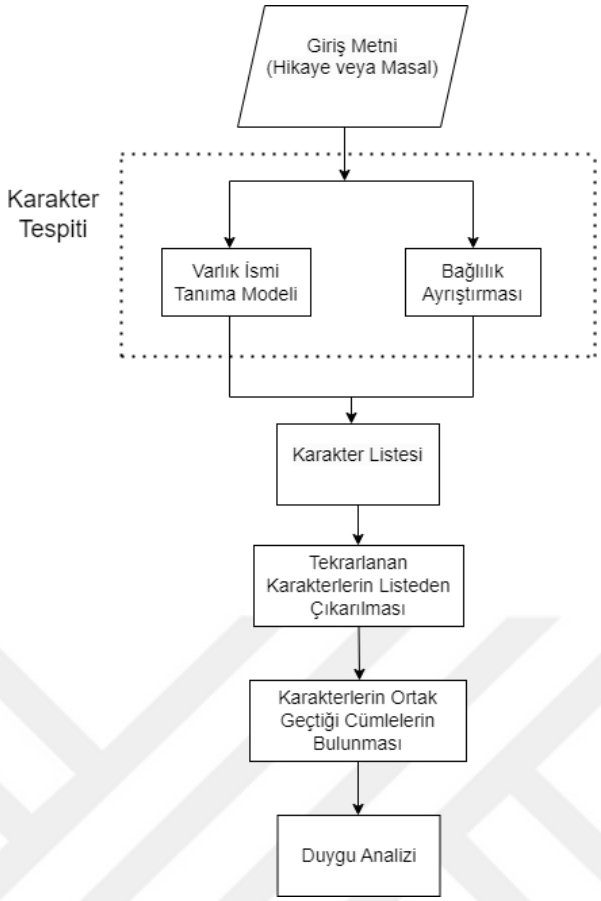


Şekil 3.15 BERT dönüştürücüsüne ait görsel

4
SİSTEM TASARIMI

Uygulama genel olarak üç modül olarak Python ile yazılmıştır. Bu modüllerden ilk ikisinde uygulamanın esas bileşenleri olan varlık ismi tanıma ve duygu analizi modelleri eğitilmiştir. Eğitimden sonra modeller sonraki aşamalarda kolay bir şekilde yüklenip kullanılması için joblib ve torch.save araçları ile dosya olarak kaydedilmiştir. Üçüncü modülde ise bu iki model birlikte bir algoritma içinde kullanılarak ana karakterlerin tespiti ve aralarındaki duygu analizi yapılmıştır. İlaveten bu dosyada girdi olarak kullanılacak olan metnin cümlelere ayrılması, gereksiz olan noktalama işaretlerinden temizlenmesi ve bağlılık ayrıştırma modelinin uygulanması gibi işlemler gerçekleştirilmiştir.

Uygulamanın genel algoritmik akışı şu şekildedir: İlk adımda, masaldaki karakterler belirlenmiştir. Bu aşama için çalışma kapsamında eğittimiz varlık ismi tanıma modeli ve doğal dil işleme projelerinde sıkca kullanılan spaCy kütüphanesinin sağladığı bağlılık ayrıştırması yöntemi kullanılmıştır. Ardından, karakterlerin birlikte geçtiği cümleler seçilmiştir. Bu cümleler bir listeye alındıktan sonra eğitilen model ile her bir cümlenin duygusal tonu tespit edilmiştir. Son olarak, basit istatistiksel yöntemler kullanılarak pozitif veya negatif yönelimli cümlelerin sayısına göre karakterler arasındaki duygusal ton belirlenmiştir (Şekil 4.1).



Şekil 4.1 Uygulamanın algoritmik akışına ait basit bir görsel

4.1 Varlık İsmi Tanıma Modeli

Çalışma kapsamında karakterlerin belirlenmesi için kullanılan varlık ismi tanıma modeli nerBert isimli ipynb uzantılı dosyada eğitilmiştir. Bu derin öğrenme modeli 26 bin cümleden oluşan veri setiyle eğitilen bir BERT dönüştürücüsüdür. Varlık ismi tanıma modelleri metin içindeki kişi, yer, organizasyon gibi varlıkların tespit edilmesi için kullanılır ve her varlığın kendi etiketi bulunmaktadır. Çalışma kapsamında sadece B-PERSON ve I-PERSON etiketli isimler seçilerek karakter listesine eklenmiştir. Burada B-PERSON ismin cümlenin başında, I-PERSON ise ortada veya sonda geldiğini belirtmektedir (Şekil 4.2).

Varlık ismi tanıma modelinin eğitiminin ilk aşamasında veri setinin eğitim ve test için bölümlere ayrılması yapılmıştır. Veri setinin yüzde sekseni eğitim yüzde yirmisi ise test için ayrılmıştır. Ardından, test ve eğitim verileri modelin giriş ve çıkış olarak kabul ettiği sütunlara ayrılarak iki ayrı veri yapısı olarak oluşturulmuştur (data frame).

Veri setinin “sentence id” ve “words” sütunları modelin girdi verisi olarak ayrılmıştır. Bu modelin eğitimi için kullanılan veri setinde her satır için bir kelime, cümlenin ”id”-si ve çıktı verisi olan etiket yer almaktadır.

```
labels
O                321773
B-PERSON          11896
B-LOCATION         8686
B-ORGANIZATION     7587
I-PERSON           5551
I-ORGANIZATION     5064
B-DATE             1250
I-LOCATION         1225
I-DATE             1078
I-MONEY            1048
I-PERCENT           640
B-PERCENT           610
B-MONEY             495
B-TIME              156
I-TIME              145
Name: count, dtype: int64
```

Şekil 4.2 Varlık ismi tanıma modelinin veri setindeki etiket sayısı bilgileri

Eğitim için kullanılan veri setinde varlık isim kategorisinde olan etiketlerden en çok B-PERSON ve B-LOCATION etiketleridir. Veri setinde uygulama için gerekli olan yaklaşık 11896 B-PERSON ve 5551 I-PERSON etiketli kelime bulunmaktadır. Eğitim döngüsü – “epok” değeri eğitim veri setinin model üzerinden kaç kere geçeceğini belirtir ve varlık ismi tanıma modeli için bu değer 10 olarak seçilmiştir. Her eğitim adımında kullanılacak veri miktarını bildiren “batch size” hem eğitim hem de test aşaması için 32 olarak belirlenmiştir. Optimizasyon işleminin hassasiyetini kontrol eden öğrenme oranı 1e-4 olarak ayarlanmıştır.

Eğitim parametreleri ayarlandıktan sonra “NERModel” sınıfı ile “bert-base-cased” isimli modelin, kullanılacak etiketler ile birlikte nesnesi oluşturulmuştur.

Model eğitim aşamasından sonra “joblib” kütüphanesi ile bilgisayara ardından da “Google Drive”-a kaydedilmiştir. Modelin test verisiyle değerlendirilmesinden sonra F1 skor değeri 0,90, duyarlılık değeri 0,91, kesinlik değeri ise 0,90 olarak belirlenmiştir.

Bir sonraki adımda model basit bir cümle ile test edilmiştir. “Günlerden bir gün Lara yine ormana yürüyüş yapmaya gitmiş” cümlesi modele girdi olarak verilmiştir.

Tahmin aşamasından sonra cümledeki kelimeler ve etiketleri sonuç olarak "predictionNer" değişkenine eklenmiştir. Ardından basit bir algoritma ile bu değişkenin içinden "B-PERSON" ve "I-PERSON" etiketli özel isimler çıkarılmıştır (Şekil 4.3).

Şekil 4.3 Varlık ismi tanıma modelinin sonuçlarından kişi isimlerinin çıkarılması

Şekil 4.4'deki kod parçasında eğitilmiş olan modelin "joblib" ile dosyaya çevrilmesi gösterilmiştir. Bu işlem sayesinde modelin kolay bir şekilde yeniden "Google Colab"-a yüklenip kullanılması mümkündür. İlaveten dosyanın "Google Drive"-a eklenmesi kod içinde dosyanın çağırılmasını daha da kolaylaştırmıştır.

```
#joblibi ice aktar
model_dosyasi_ismi = 'trained_modelCud.joblib'
joblib.load(model, model_dosyasi_ismi)
```

Şekil 4.4 Varlık ismi tanıma modelinin joblib ile kaydedilmesine ait kod

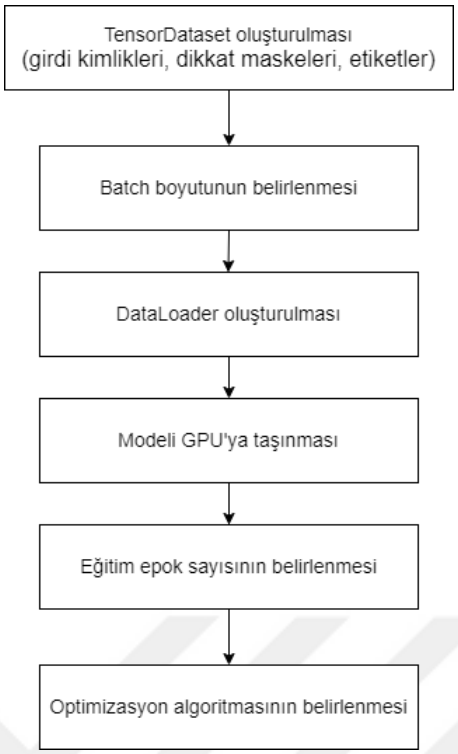
4.2 Duygu Analizi Modeli

Duygu analizi modelinin eğitimi “sentBert” isimli “ipynb” uzantılı dosyada yapılmıştır. Varlık ismi tanıma modeli gibi duygu analizi için kullanılan model de BERT dönüştürücüsüdür. Yaklaşık 30 bin cümleden oluşan veri setiyle eğitilen bu model çıktı olarak cümlenin duygusal tonu pozitifse “1”, negatifse “0” döndürmektedir. Veri setinde toplam 19582 pozitif, 11932 negatif eğilimli cümle bulunmaktadır. Veri setinin yüzde sekseni eğitim, yüzde yirmisi ise modelin testi için kullanılmıştır.

Toplam 25212 cümle eğitim, 6126 cümle ise test için kullanılmıştır. Bir sonraki aşamada BERT dönüştürücüsünün eğitim aşamasında kullanması için gerekli olan “girdi kimlikleri”, “dikkat maskeleri” ve “etiketler” liste değişkenleri üretilmiştir. Cümleler tokenlere ayrıldıktan sonra her kelimenin token kimliği “girdi kimlikleri”, dikkat maske değeri ise “dikkat maskeleri” liste değişkeninde tutulmaktadır. Ardından, bu listeler “torch” ile tensorlara dönüştürülüp birleştirilmiştir. Eğitim etiketlerini saklayan “labels” listesi de aynı şekilde tensora dönüştürülmüştür.

Bu üç tensor BERT modelinin eğitimi için kullanılacak olan “TensorDataset” değişkenini oluşturmaktadır. Eğitim aşaması için "batch" ölçüsü 32 olarak belirlenmiştir. Bir sonraki aşamada “DataLoader” nesnesi oluşturulup önceden hazırlanan “TensorDataset – train dataset” isimli veri setinden rastgele “batch” ölçüsü miktarında örnek veriler hazırlanmıştır. Ardından, "dbmdz/bert-base-turkish-128k-uncased" isimli BERT modeli yüklenmiştir. Bir sonraki aşamada model duygu analizi – sınıflandırma görevi için pozitif ve negatif sınıfı ayırt edebilecek şekilde yukarıda belirtilen veri seti ile yeniden eğitilmiştir. Duygu analizi modeli için “epoch” değeri 5 seçilmiştir. Eğitim aşamasında sinir ağının katmanlarındaki ağırlıkları optimize etmek için “AdamW” metodu seçilmiştir. Öğrenme oranı 5e-5, öğrenmenin hızını kontrol eden epsilon ise 1e-8 olarak ayarlanmıştır (Şekil 4.5).

Hem varlık ismi tanıma hem de duygu analizi modeli “Google Colab” platformunda GPU üzerinde (paralel ve hızlı hesaplama seçimi) “Cuda” ile eğitilmiştir.



Şekil 4.5 Duygu analizi modelinin eğitim sürecinin adımlarına ait diagram

Duygu analizi modeli eğitildikten sonra Pytorch kütüphanesinin "torch.save" metodu ile dosya formatına dönüştürülüp bilgisayara kaydedilmiştir.

Bir sonraki aşamada modelin test veri seti ile F1, duyarlılık ve kesinlik skorları tespit edilmiştir. Benzer olarak "girdi kimlikleri", "dikkat maskeleri" değerleri bu sefer test veri seti için "tokenizer.encode plus" metodu ile bulunup listeye eklendikten sonra tensöre dönüştürülmüştür. Ardından, bu tensörleri içeren bir "tahmin veri kumesi" isimli "TensorDataset" nesnesi oluşturulmuştur. Test için kullanılan bu verilerin sıralı olarak seçilmesi için "SequentialSampler" nesnesi yaratılmıştır. "DataLoader" sınıfı kullanılarak, "prediction data" verisini "batch" ölçüsüne uygun olarak sıralı bir şekilde çağıran bir veri yükleyici oluşturulmuştur.

Uygulamanın tahminlerini ve gerçek etiketleri saklayan listeler oluşturulduktan sonra her "batch" için döngü içinde tahminler yapılmıştır. Modelin test verisi için tahmin işlemi kodda "with torch.no grad()" bloğu içinde gerçekleştirilmiştir. Bu tahminler birer "numpy" dizisine dönüştürüldükten sonra "predictions" isimli listeye eklenmiştir.

Duygu analizi modelinin test veri setiyle değerlendirme aşamasından sonra F1, duyarlılık ve kesinlik skor değerleri 0.98 olarak belirlenmiştir.

Modelin tahmin aşamasından sonra sınıflar için hesaplanan ham tahmin skorları olan “logits” değerleri için “softmax” fonksiyonu uygulanmıştır. Bu fonksiyondan geçtikden sonra bu skorlar iki sınıf için toplamı 1 olacak şekilde bir olasılık değeri almışlardır. Bu olasılık değerleri “probabilities” değişkeninin içine eklenmiştir. Son olarak “probabilities” değişkenine “argmax” fonksiyonu uygulanarak modelin son çıktı değeri elde edilmiştir.

4.3 Modellerin Birlikte Kullanımı

Varlık ismi tanıma ve duygu analizi modellerinin eğitimi tamamlanıp kayıt ettikten sonra bu model dosyaları çalışmanın üçüncü “sentNerAlgo” isimli bir ipynb dosyasında bir arada kullanılmıştır. İlaveten spaCy kütüphanesinin bağlılık ayrıştırma modeli de uygulanarak girdi olarak verilen masallardan karakterler çıkarılıp duygu analizi yapılmıştır. Varlık ismi tanıma modeli, masal karakterleri olan isimlerin ilk harfinin büyük yazıldığı pek çok metin için doğru çalışsa da, bazı masallar için doğru sonuçlar vermemektedir. Bunun sebebi masallardaki karakterlerin özel isimler yerine, küçük harfle yazılmış cins isimlerle kullanılıyor olmasıdır. Örneğin, “Kırmızı Başlıklı Kız” masalında karakterlerin isimleri “kız”,” büyükanne”,” kurt” gibi küçük harfle yazılmış genel isimlerle ifade edilmektedir. Bu durumda, masaldaki karakterlerin tespit edilmesi için ilaveten bağlılık ayrıştırma yönteminin kullanılmasına karar verilmiştir. Bağlılık ayrıştırması modeli cümlede kelimeler arasındaki bağlamsal ilişkiyi belirlemek için kullanılır. Bu yöntemle kelimeler arasındaki ilişkiler bir ağaç yapısında etiketler ile temsil edilir. Uygulamada cümle içinden sadece “nsubj” ve ”ROOT” etiketli özneler çıkarılmıştır. Çalışmada kullanılan bağlılık ayrıştırma modeli spaCy kütüphanesinin bir aracıdır. Uygulamanın genel olarak ilk aşaması olan karakter tespitinde bu iki model paralel olarak birlikte kullanılmıştır.

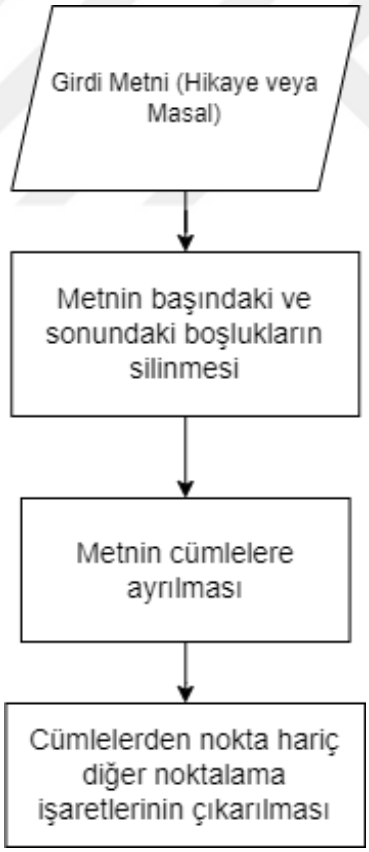
Bu “colab” dosyasında ilk adımda eğitimden sonra dosya formatında kayedilen varlık ismi tanıma modeli “joblib.load”, duygu analizi modeli ise “torch.load” metotları ile kullanıma hazır olarak “Google Drive”-dan dosya yolu belirtilerek “colab” dosyasına yüklenmiştir. “Joblib” kütüphanesi modelin kendisini direkt olarak yükleyebilmektedir. “Torch” kütüphanesi ile yükleme zamanı, önce modelin yapısı tanımlanmış, ardından hazır ağırlık değerleri “Google” sürücüsünden

yüklenmiştir. Sürücüde varlık ismi tanıma modeli “trained modelCud”, duygu analizi modeli ise “depthsent2Cud” olarak kaydedilmiştir.

Bağlılık ayrıştırma modeli ise “spaCy” kütüphanesi “import” edildikten (içe aktarıldıktan) sonra “spacy.load” metodu ile çağırılmıştır. Model kütüphanede “tr core news trf” ismiyle bulunmaktadır.

Uygulama girdi olarak “txt” dosyasında bulunan masal metnini kabul etmektedir. İlk adımda girdi metni ile karakter tespiti aşamasından önce bazı Python metotları ile ektra boşluklardan ve gereksiz noktalama işaretlerinden temizlenmiştir. Ekstra boşluklara misal olarak, metnin en başında ve en sonunda bulunan veya nokta işaretinden sonra gelen çift boşluklar gösterilebilir. Bu metodun içinde düzenli ifade deseni (“regex”) kullanılmıştır (Şekil 4.6).

Noktadan başka diğer noktalama işaretlerinin silinmesi için de boşlukların temizlenmesinde olduğu gibi “regex” kullanılmıştır.



Şekil 4.6 Metin üzerinde yapılan bazı önişlemlerin akışı

Uygulamanın testi için metin olarak kullanılan masalların içinde meşhur olan "Kırmızı Başlıklı Kız" masalı da yer almaktadır. Metin "file.read" komutu tarafından okunduktan sonra yukarıda belirtilen fonksiyon ile ektra boşluklardan temizlenmiştir. Ardından metin "regex" kullanılıp nokta işaretleri dikkate alınarak cümlelere ayrılmış ve bu cümleler "sentences" isimli listeye eklenmiştir. Bu listedeki cümlelerde, döngü içinde virgül, iki nokta veya noktalı virgül gibi işaretler silindikten sonra karakter tespiti için kullanılan modeller çağırılmıştır. Varlık ismi tanıma modeli "varliklariAl(s)", bağlılık ayrıştırma modeli ise "varliklariAlBaglilikAyristirma(s)" (Şekil 4.7) fonksiyonunda çalışmaktadır.

Her iki fonksiyonun içinde modellerin tahmin aşaması tamamlandıktan sonra karakter olarak tespit edilen özel isimler iki farklı listeye eklenmiştir. Varlık ismi tanıma modelinin çıktılarının eklendiği liste "kisi listesi", bağlılık ayrıştırma modelinin çıktılarının eklendiği liste ise "kisi listesi baglilik ayristirma" olarak isimlendirilmiştir. Sonraki adımların birinde bu iki liste isimler tekrarlanmayacak şekilde birleştirilmiştir.

```
// Baglilik ayristirma modelinin ciktilarinin listeye
   eklenmesi
def varliklari_al_baglilik_ayristirma(cumle):
   dokuman = nlp1(cumle)
   dongu icinde dokuman:
      if ('nsubj' cumle.dep_ icinde) veya ('ROOT'
         cumle.dep_ icinde):
         kisi_listesi_baglilik_ayristirma.append(str(cumle.lemma_))
```

Şekil 4.7 Bağlılık ayrıştırması modelinin çıktılarının listeye eklenmesine ait sözde kod

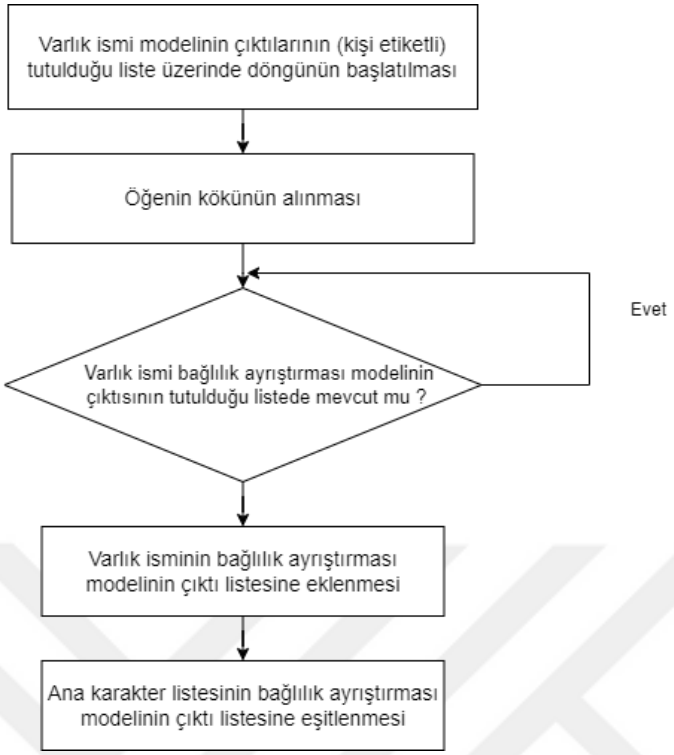
Bağlılık ayrıştırma modeli genel özneleri tespit ettiğinden dolayı özel karakter isimlerinden başka genel anlamlı özneler de "kisi listesi baglilik ayristirma" isimli listeye eklenmiş olabilir. Örnek olarak, bu model masal içinde geçen "ev" kelimesini de "nsubj" etiketli isim olarak algıladığından dolayı yanlışlıkla listeye ekleyebilir. Bu sebepten basit bir koşul eklenerek metin içinde belirli bir sayının üstünde kullanılan özneler karakter ismi olarak seçilmiştir. Bu sayı masal içindeki cümlelerin sayısı ile belirlenmektedir. Uygulamanın genel olarak testi için

kullanılan masallar ortalama 80 ile 120 arası sayıda cümleden oluşmaktadır. Özneleri tespit etmek için kullanılan sayı, cümle sayısının yüzden az olduğu durumlarda 8, yüzden çok olduğu durumlarda ise 12 olarak belirlenmiştir. Bu basit metot uygulamanın gelecekte daha verimli ve kesin çalışması için geliştirilmesi planlanan bölümlerinden biridir.

Karakter tespiti aşaması her iki modelde döngü içinde, metin içindeki her cümle için ayrılıkta yapılmaktadır. Bu sebepten, model bazı durumlarda bir karakter ismini birden fazla cümle içinde bulabilir. Bu durumda, bir varlık ismi çıktı listesine birden fazla sayıda eklenebilir. Örnek olarak, varlık ismi tanıma modelinin çıktılarının bulunduğu listeye "Ahmet" ismi iki kere eklenmiş olabilir. Aynı şekilde bağlılık ayrıştırması modelinin bulduğu isimlerden oluşan listede "kurt" karakter ismi birden fazla sayıda bulunabilir. Bu durumun engellenmesi için her iki liste "set" (tekrarlanmayan öğelerin tutulduğu liste) veri yapısına dönüştürülüp yeniden listeye çevrilmiştir (Şekil 4.8).

Şekil 4.8 Listede tekrarlanan isimlerin silinmesinin basit akış diagramı

Bir sonraki aşamada her iki listede bulunan karakter isimleri tekrarlanmayacak şekilde tek bir karakter listesine toplanmıştır. İlk olarak, varlık ismi tanıma modelinin çıktılarını saklayan listedeki her eleman için döngü başlatılmıştır. Eğer bu eleman bağlılık ayrıştırma modelinin çıktılarının bulunduğu listede bulunmuyorsa bu listeye eklenmektedir. Bununla da her iki modelin çıktıları tekrarlanmayacak şekilde "kisi listesi baglilik ayristirma" listesine eklenmiştir (Şekil 4.9).



Şekil 4.9 Varlık ismi tanıma ve bağlılık ayrıştırması modellerinin çıktı listelerinin birleştirilmesine ait akış diagramı

İsimlerin bu listedeki varlığının kontrolü aşamasından önce "Zemberek" kütüphanesinin bir aracı olan "Zeyrek" ile döngüde sırası gelen elemanın kökü tespit edilmiştir. Bunun sebebi varlık ismi tanıma modelinin bazı modellerde karakterleri ekleri ile birlikte bulup listeye eklemesidir. Örnek olarak, listede "Mehmet" ismi "Mehmetin" şeklinde bulunuyorsa bu ismi bağlılık ayrıştırma modelinin çıktılarının tutulduğu listedeki isimlerle kıyaslayıp karakter listesine eklemek hatalı sonuçlar verebilir. Bağlılık ayrıştırması modeli genelde öznelerin köklerini çıktı olarak verdiği için listede "Mehmetin" ismi bulunmayacaktır ve bu durumda eğer bu isim ek ile birlikte sonuç listesine eklenirse, listede hem "Mehmet" hem de "Mehmetin" isimleri yer alacaktır. "Zeyrek" aracının "lemmatize" fonksiyonu bu hatayı engellemektedir.

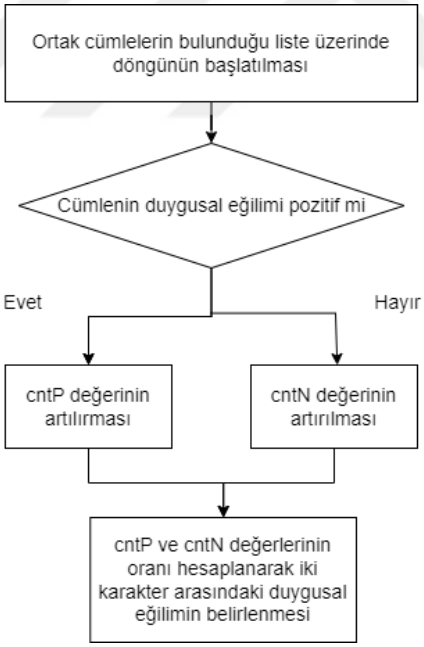
Bir sonraki aşamada bu karakterlerin ortak geçtiği cümlelerin tespit edilmesi yapılmıştır. Bu işlem için öncelikle "anaKarakterListesi" listesindeki karakterler ikişerli gruplar halinde iki boyutlu bir listeye toplanmışlardır (Şekil 4.10).

```
#ornek [['kurt', 'kiz'],['kiz', 'babaanne'],['kurt',
   'babaanne']]
```

Şekil 4.10 İki boyutlu karakter listesine ait görüntü

Sonraki aşamada, yeni bir döngü içinde bu iki boyutlu liste içindeki her elemanın kapsadığı isimlerin ortak geçtiği cümleler tespit edilmiştir. Bu cümleler, "ortakCumleler" isimli listeye eklenmiştir. Karakter isimlerini içeren cümlelerin bulunmasında "regex" deseni uygulanmıştır.

Son aşamada basit bir istatistiksel metot ile karşılaştırılan iki masal karakterinin arasındaki duygusal ilişki belirlenir. "ortakCumleler" isimli listede bulunan cümlelerin duygusal tonu önceden eğitilmiş olan duygu analizi modeli ile tespit edilmiştir. Duygu analizi modeli cümlenin duygusal tonunu pozitif belirledikte çıktı olarak 1 döndürmektedir. Bu durum tekrarlandığında "cntP" değişkeninin değeri artmaktadır. Cümle negatif duygulu ise modelin çıktısı 0 olup, "cntN" değişkeninin değerini artırmaktadır (Şekil 4.11).



Şekil 4.11 Duygu analizi modelinin uygulanmasının akışı

5
DENEYSEL SONUÇLAR

Geliştirilen uygulamanın genel başarısını tespit edebilmek için 21 farklı masal ile test yapılmıştır. Çocuklar için yazılmış olan bu masallar farklı web sitelerinden alınmıştır. Masal Metinleri'de örnek olarak Lustig masalının içeriği, masalda yer alan kahramanlar verilmektedir. Tablo 5.1'de her bir masalın adı, masaldaki toplam cümle sayısı ve her masalın içerisinde yer alan karakterlerinin sayısı verilmektedir. Masallarda yer alan karakterler ve karakterler arasındaki ilişkiler iki farklı kişi tarafından etiketlenmiştir.

Tablo 5.1 Test verisinde yer alan masallar ve sahip olduğu özellikler

Masalın Adı	**Toplam Cümle Sayısı**	**Toplam Kahraman Sayısı**
Zehra	103	4
Hans Efendi	62	4
Sarman	86	5
Vezir Nureddin	121	3
Uyuyan Güzel	80	5
Rapunzel	74	3
Pinokyo	183	4
Pamuk Prenses	94	4
Lustig	62	3
Kurbağa Henry	74	3
Kırmızı Başlıklı Kız	52	3

Tablo 5.1 Test verisinde yer alan masallar ve sahip olduğu özellikler (devamı)

Kınalı Kuzu	77	4
Kamer	115	4
Hansel ve Gretel	211	4
Kağıt	146	3
Güzel ve Çirkin	166	3
Eşek ve Öküz	56	4
Habil ve Kabil	163	4
Çirkin Prens	87	5
Ceviz	137	4
Balıkçı	97	4

Denemeler yapılırken ilk olarak her masalda yer alan birinci ve varsa ikinci karakterler tespit edilmiştir. Bir sonraki aşamada da diğer karakterlerin tespit edilmesindeki başarı kriteri ölçülmüştür. Masallarda yer alan karakterler tespit edildikten sonra, karakterler arasında oluşturulan ikili eşleşmeler çıkarılarak cümlelerde aranıp, her cümlenin karakterler açısından duygu durumu tespit edilerek masal içinde en baskın olan duygu durumu karakter çiftlerine atanmıştır. Cümle sayısı fazla olan uzun masallarda karakterler arasındaki duygu durumu tespit edilirken masalın giriş, gelişme ve sonuç bölümlerinde duygu durumunun değişip değişmediği kontrol edilerek okuyucuya sunulmuştur. Bunu yapmanın amacı bazı masallarda metnin başında karakterler arasında negatif veya pozitif olarak başlayan bir ilişki durumu masalın ilerleyen bölümlerinde değişebilmektedir.

Masallardaki karakterlerin sayısı çoğunlukla üç olarak belirlenmiştir. Bu karakterler değerlendirme tablolarında "Birinci Karakter", "İkinci Karakter" ve "Diğer Karakterler" olarak isimlendirilmektedir. Tablo 5.1'in üçüncü kolonunda her masal için işaretlenmiş olan karakterlerin sayısı verilmektedir. Masal içerisinde belli bir sayının altında kalan karakterler etiketleme yapan kişi tarafından

işaretlenmemiş ve karakter listesine alınmamıştır. Bu sıklık değeri çalışmada kullanılan masalların cümle sayısına bağlı olarak tespit edilmiştir.

Uygulamayı çalıştırdığımızda masallarda en fazla alınan ilk iki kahramanın tespit edilmesindeki başarı Tablo 5.2'de gösterilmektedir. Tablo 5.2'de "+" sembolü bize ilgili karakterin tespit edildiğini söylerken "-" sembolü o karakterin tespit edilemediği bilgisini vermektedir. Örneğin, "Kırmızı Başlıklı Kız" masalındaki iki ana karakter kız ve kurttur, üçüncü ve bizim için diğer karakter olan isim de büyük annedir. Uygulama bu masal için ilk iki karakteri tespit edebilmiştir. Buna karşılık "Hansel ve Gretel" masalında ikinci ana karakter tespit edilememiştir. Yirmibir masalın tamamında ilk karakterlerin 100% tespit edilmiş, ikinci karakterlerin ise 15/21 tespit edilmiş olup, başarı 71% olmuştur.

Tablo 5.2 Varlık ismi tanıma modeli ilk iki karakter tespiti

Masalın Adı	**1. Karakter**	**2. Karakter**
Zehra	+	-
Hans Efendi	+	-
Sarman	+	+
Vezir Nureddin	+	+
Uyuyan Güzel	+	-
Rapunzel	+	+
Pinokyo	+	+
Pamuk Prenses	+	+
Lustig	+	+
Kurbağa Henry	+	+
Kırmızı Başlıklı Kız	+	+

Tablo 5.2 Varlık ismi tanıma modeli ilk iki karakter tespiti (devamı)

Kınalı Kuzu	+	+
Kamer	+	+
Hansel ve Gretel	+	-
Kağıt	+	-
Güzel ve Çirkin	+	+
Eşek ve Öküz	+	+
Habil ve Kabil	+	+
Çirkin Prens	+	+
Ceviz	+	+
Balıkçı	+	-

Masal içinde ikinci ana karakterin bulunamama sebeplerinden biri, bazı metinlerde karakter isminin lemmatizasyon aşamasında orijinal formunu kaybetmesidir. Örnek olarak, Hansel ve Gretel masalında uygulama, Hansel karakterini bulmasına rağmen Gretel'i bulamamıştır. Çünkü lemmatizasyon sürecinde "Gretel" kelimesindeki "el" ek olarak algılanmış ve silinmiştir, bu yüzden karakter "Gret" olarak tespit edilmiştir ve tanınamamıştır.

Bir diğer örnek ise "Zehra" isimli masaldır. Bu masalda, Zehra'ya destek olan ve aralarındaki ilişki pozitif olan ikinci karakter Cahide, Zehra'nın annesidir. Ancak, metindeki cümlelerin büyük çoğunluğunda "annesi" veya "Zehranın annesi" gibi genel isimlerle ifade edildiği için uygulama tarafından tespit edilememiştir.

Diğer karakterlerin tespitinde ise 38 karakterin sadece 27 tanesini bulunmuştur. Diğer karakterlerin başarısı 71% olmuştur. Tüm masallar içerisinde toplamda etiketlenmiş 78 farklı karakter olup, 63 tanesi tespit edilerek 78% başarı elde edilmiştir. Tablo 5.3'de "Diğer karakterler" sütununda yer alan bilgi kaç adet diğer karakter olduğu ve kaç tanesinin tespit edildiğini vermektedir. "Toplam" sütunun da ise tüm masalda yer alan karakterlerin bu aşamaya kadar kaç tanesinin tespit

edildiğini göstermektedir. Örneğin “Pinokyo” masalında toplamda 4 kahraman vardır. Bu kahramanların ilk iki ana kahramanı bulunmuş, diğer iki kahramanın da Tablo 5.3’de “Diğer karakterler” sütununda sadece bir tanesinin tespit edildiği görülmektedir. Bu sebeple “Toplam” sütununda 4 karakterin sadece 3 tanesinin tespit edildiği bilgisi yer almaktadır.

Tablo 5.3 Varlık ismi tanıma modeli diğer karakterler tespiti

Masalın Adı	**Diğer Karakterler**	**Toplam**
Zehra	2/2	3/4
Hans Efendi	1/2	2/4
Sarman	2/3	4/5
Vezir Nureddin	1/1	3/3
Uyuyan Güzel	2/3	3/5
Rapunzel	1/1	3/3
Pinokyo	1/2	3/4
Pamuk Prenses	1/2	3/4
Lustig	0/1	2/3
Kurbağa Henry	1/1	3/3
Kırmızı Başlıklı Kız	1/1	3/3
Kınalı Kuzu	2/2	4/4
Kamer	0/2	2/4
Hansel ve Gretel	2/2	3/4
Kağıt	0/1	1/3

Tablo 5.3 Varlık ismi tanıma modeli diğer karakterler tespiti (devamı)

Güzel ve Çirkin	1/1	3/3
Eşek ve Öküz	0/2	2/4
Habil ve Kabil	2/2	4/4
Çirkin Prens	3/3	5/5
Ceviz	2/2	4/4
Balıkçı	2/2	3/4

Tablo 5.4'de herbir masal için çıkarılan kesinlik, duyarlık ve F1 skorları verilmiştir. Kesinlik, bulduğu karakterlerin gerçek olan karakterlere oranıdır. Bir başka deyişle Eşitlik 5.1'de verildiği gibi Doğru Pozitiflerin (DP), Doğru Pozitif ve Yanlış Pozitiflerin (YP) toplamına oranıdır.

$$\text{Kesinlik} = \frac{DP}{DP + YP} \tag{5.1}$$

Duyarlılık ise, olması gereken karakterlerin kaç tanesinin getirildiği yani Eşitlik 5.2'de verildiği üzere Doğru Pozitiflerin, Doğru Pozitif ve Yanlış Negatiflerin (YN) toplamına oranıdır.

$$\text{Duyarlılık} = \frac{DP}{DP + YN} \tag{5.2}$$

F1 skor ise bu iki değerin harmonik ortalaması olup, Eşitlik 5.3'deki gibi hesaplanır.

$$\text{F1 Skoru} = \frac{2 \times Kesinlik \times Duyarlılık}{Kesinlik + Duyarlılık} \tag{5.3}$$

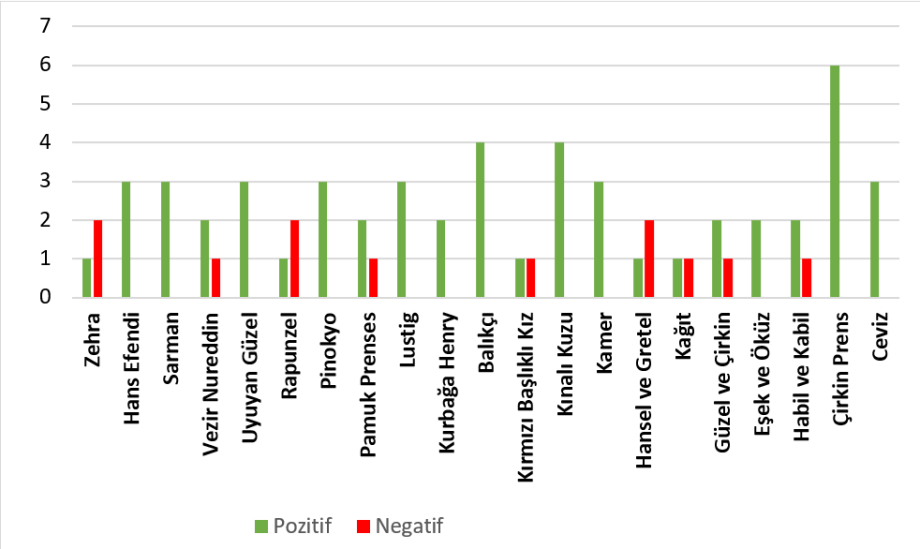
Tablo 5.4 Masallardaki karakterlerin çıkarım başarısı

Masalın Adı	**Kesinlik**	**Duyarlılık**	**F1 Skor**
Zehra	0,75	0,75	0,75
Hans Efendi	0,66	0,50	0,57

Tablo 5.4 Masallardaki karakterlerin çıkarım başarısı (devamı)

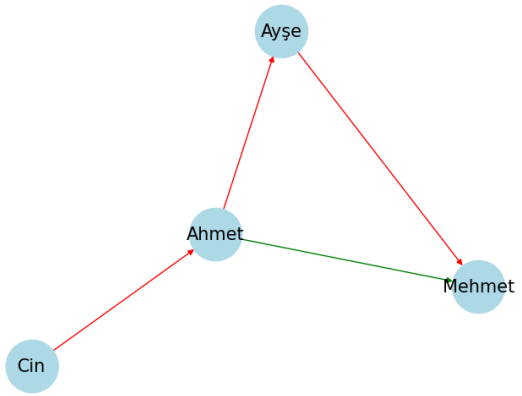
Sarman	0,80	1	0,88
Vezir Nureddin	0,60	1	0,75
Uyuyan Güzel	1	0,6	0,75
Rapunzel	0,6	1	0,75
Pinokyo	0,5	0,75	0,6
Pamuk Prenses	0,6	0,75	0,6
Lustig	0,4	0,66	0,5
Kurbağa Henry	0,75	1	0,86
Kırmızı Başlıklı Kız	1	1	1
Kınalı Kuzu	0,80	1	0,89
Kamer	0,67	0,50	0,57
Hansel ve Gretel	0,50	0,75	0,60
Kağıt	0,33	0,33	0,33
Güzel ve Çirkin	1	1	1
Eşek ve Öküz	0,40	0,50	0,44
Habil ve Kabil	1	1	1
Çirkin Prens	1	1	1
Ceviz	1	1	1
Balıkçı	1	0,75	0,86

Çalışmanın ikinci aşamasında tespit edilen karakterler ve aralarındaki duygusal ilişkinin çıkarılması için karakterlerin geçtiği ortak cümleler belirlenmiş ve duygu analizi modeline verilmiştir. Bu aşamada tespit edilemeyen karakterlerin diğer masal karakterleri ile ilişkileri de tespit edilememiştir. Masallar çocuklar için yazılmış olduğundan karakterler arasındaki ilişkiler genellikle pozitif yönde olmaktadır. Yirmibir masalın içinde elle yapılan etiketleme sonucunda Şekil 5.1'de gösterildiği üzere toplamda 52 pozitif, 12 de negatif ilişki tespit edilmiştir.

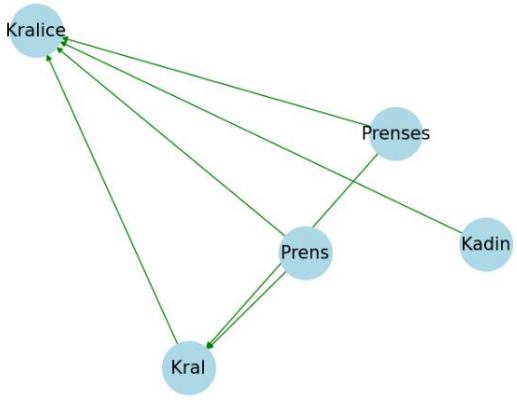


Şekil 5.1 Masal kahramanları arasındaki duygusal ilişkinin sayısı

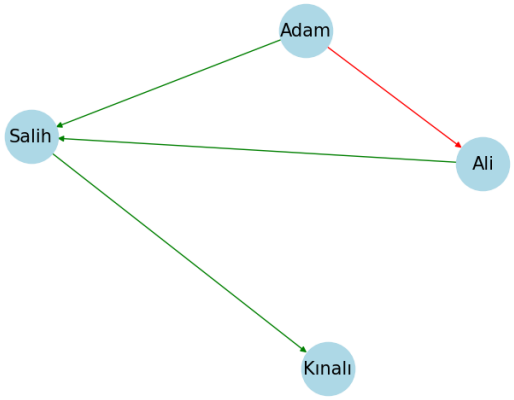
Uygulamada tespit edilen ilişkiler Pythonun sıkça kullanılan Matplotlib kütüphanesi ile görselleştirilmiştir. Şekil 5.2, 5.3, 5.4'de sırasıyla uygulamanın Balıkçı, Çirkin Prens ve Kınalı Kuzu masalları için ikili karakterler arasındaki ilişkinin masalın genelindeki duygu durumları çizge graflar ile gösterilmektedir.



Şekil 5.2 Balıkçı masalının tamamına ait ilişki diagramı



Şekil 5.3 Çirkin Prens masalının tamamına ait ilişki diagramı



Şekil 5.4 Kınalı Kuzu masalının tamamına ait ilişki diagramı

Model tarafından kahramanların ikili olarak geçtiği cümlelerin duygu durumları tespit edilip değerlendirildiğinde, tüm masalların genelinde 52 pozitif ilişkinin 24, 12 negatif ilişkinin de 8'si tespit edilebilmiştir. Modelin pozitif ilişkileri bulma başarısı ortalama 46% iken, negatif ilişkileri bulma başarısı da 66% olmuştur. Tablo 5.5'de her bir masal için bulunan ilişkilerin doğruluk oranları gösterilmektedir. Negatif ilişkilerin sayısı daha az olsa bile, tespit edilme başarısı pozitif ilişkilere daha yüksektir.

Tablo 5.5 Duygu analizi modelinin başarısı

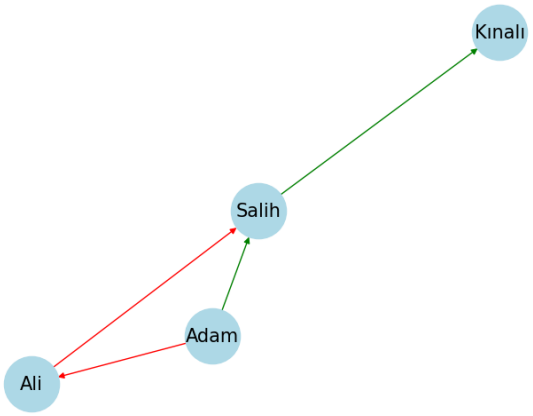
Masalın Adı	**Pozitif İlişki**	**Negatif İlişki**
Zehra	0/1	1/2
Hans Efendi	1/3	-

Tablo 5.5 Duygu analizi modelinin başarısı **(**devamı**)**

Sarman	1/3	-
Vezir Nureddin	1/2	1/1
Uyuyan Güzel	2/3	-
Rapunzel	1/1	2/2
Pinokyo	1/3	-
Pamuk Prenses	0/2	0/1
Lustig	1/3	-
Kurbağa Henry	0/2	-
Kırmızı Başlıklı Kız	0/1	1/1
Kınalı Kuzu	3/4	-
Kamer	2/3	-
Hansel ve Gretel	0/1	1/2
Kağıt	0/1	0/1
Güzel ve Çirkin	2/2	1/1
Eşek ve Öküz	0/2	-
Habil ve Kabil	0/2	1/1
Çirkin Prens	6/6	-
Ceviz	2/3	-
Balıkçı	1/4	-

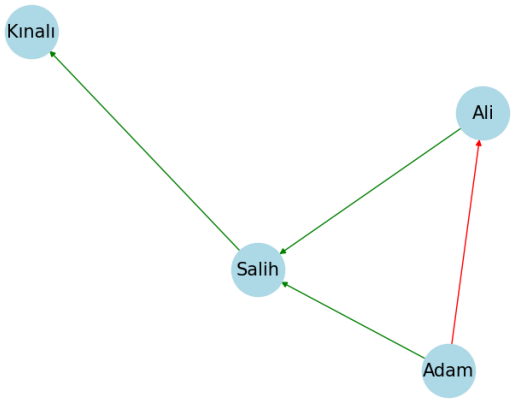
Masalların genelinde duygusal ilişkilerin tespitinde Kesinlik 76%, Duyarlılık 50% ve F1 skor'da 60% olarak elde edilmiştir. Karakterler arasındaki duygu değişimini çok kısa olan masallarda göstermek güç olacağı için, cümle sayısı fazla olan nispeten diğer masallar göre daha uzun olan dört masal seçilmiştir. Bu masallar "Hansel ve Gretel", "Kırmızı Başlıklı Kız", Kınalı Kuzu" ve "Lustig" olmuştur. Duygu Analizi modelini çalıştırdığımızda ilk masalımız olan "Hansel ve Gretel" de başlangıçta cadı ile Hansel arasındaki pozitif olan ilişki, masalın sonunda negatife dönüyor olsa da bu durum tespit edilememiştir. İkinci masalımız "Kırmızı Başlıklı Kız"'da, başlangıçta pozitif başlayan kız ile kurt arasındaki ilişki (kurt, ilk başta kendini kıza iyi biri gibi göstermektedir) sonradan negatife dönmüş ve bu durum uygulama tarafından başarılı bir şekilde tespit edilmiştir. Üçüncü masalımız "Kınalı Kuzu" da Ali ve Salih arasında başlangıçta negatif olup, sonradan pozitife dönen ilişki de uygulama tarafından tespit edilmiştir. Son masalımız "Lustig" de Aziz Petrus ile Lustig arasındaki önce negatif, sonrasında pozitife dönen ilişki değişimi de uygulama tarafından tespit edilememiştir.

Kınalı Kuzu masalında 4 önemli karakterin hepsi uygulama tarafından tespit edilebilmiştir. Bu karakterlerden Ali ile Adam arasındaki ilişki metnin başında ve sonunda pozitif olmuştur. Adam isimli karakter ile Kınalı (kuzu) arasında bir ilişki yoktur. Ali ile Salih arasındaki ilişki ise metnin başında negatif olarak belirlenmiştir (Şekil 5.5).



Şekil 5.5 Kınalı Kuzu masalının giriş bölümündeki duyguya ait ilişki diagramı

Masalın sonuna doğru ise bu iki karakter arasındaki ilişki pozitif yönümlü değişmiştir (Şekil 5.6).



Şekil 5.6 Kınalı Kuzu masalının gelişme bölümünde duyguya ait ilişki diagramı

6
SONUÇ

Çalışma kapsamında, Türkçe yazılmış masallardaki karakterlerin tespit edilmesi ve bu karakterler arasındaki duygusal eğilimin belirlenmesini sağlayan bir uygulamanın geliştirilmesine odaklanılmıştır. Uygulama tarafından ilk aşamada "spaCy" kütüphanesinin sağladığı ve Türkçe için mevcut olan bağlılık ayrıştırma analizi modeli ve bizim tarafımızdan eğitilen varlık ismi tanıma modeli ile girdi olarak verilen masallardaki karakterlerin tespiti yapılmıştır. Bu aşamada bağlılık ayrıştırma modelinin kullanılmasının sebebi varlık ismi tanıma ile tespit edilemeyen kişilerin ya da masalda canlı bir karakter olarak yer alan isimlerin tespit edilebilmesi için kullanılmıştır. Varlık ismi tanıma modelleri genellikle ilk harfi büyük yazılan özel isimleri ifade eden karakterlerin tespitinde başarılı olsa da, cins isimlerle adlandırılmış olan karakterler için doğru çalışmamaktadır. Örneğin "Kırmızı Başlıklı Kız" masalındaki ana karakterlerin isimleri için kurt, kız ve babaanne isimleri kullanılmaktadır. Bu isimlerin varlık ismi tanıma modeli ile tespiti güçtür. Bağlılık ayrıştırma modeli cümleleri biçimbilimsel analiz uyguladığı için kelimeler arasındaki ilişkiler belli etiketler ile kullanıcıya sunulduğundan özne olabilecek isimler tespit edilebilmektedir. Bir sonraki aşamada, tespit edilen karakterlerin ikili olarak birlikte bulundukları cümleler tespit edilip, bu cümlelerin duygusal eğilimleri duygu analizi modeli ile tahmin edilmiştir. Ardından çoğunluk olan duygusal ton seçilmiş ve karakterler arasındaki duygusal eğilim belirlenmiştir.

Geliştirilen uygulamanın testi için 52–221 arası cümleden oluşan ve farklı internet sitelerinden alınmış 21 masal seçilmiştir. Masal karakterlerin tespitinde ortalama yüzde 78 başarı elde edilmişken duygu tespitinde, negatif ilişkilerin çıkarılmasında pozitif ilişkilere göre daha yüksek başarı alınmıştır.

İleri de sisteme derin öğrenme tekniği ile çalışan zamir çözümlemenin de eklenmesi ile masallarda karakter ismi yerine kullanılan "o" zamiri ile ilişkili karakterler de kolaylıkla tespit edileceğinden hem karakterlerin tespiti hem de karakterlerin yer aldığı cümlelerinin seçilip duygu analizine verilmesi ile alınan başarı artacaktır. Şu an uygulamada olan zamir çözümleme kural tabanlı olarak çalışmaktadır.

Uygulamaya eklenebilecek bir diğer özellik ise masal içindeki karakterler arasındaki nötr olan ilişkinin tespit edilmesidir.

Bu çalışmanın sonucu, çocuk masallarının yazım diline daha özen gösterilmesine yardımcı olacağı gibi edebiyat alanında araştırma yapan ögrenci veya araştırmacılar tarafından da edebi eserlerin incelenmesinde, haber metinlerinin veya sosyolojik içerikli metinlerin analiz edilmesinde de kullanılabilir.

KAYNAKÇA

[1] M. Fernandez, M. Peterson, and B. Ulmer, "Extracting Social Network from Literature to Predict Antagonist and Protagonist", 2015. Erişim Tarihi: 20 Mayıs 2024. [Çevrimiçi]. Mevcut: https://nlp.stanford.edu/courses/cs224n/2015/reports/14.pdf.

[2] V. Devisree, P. R. Raj, "A Hybrid Approach to Relationship Extraction from Stories", Procedia Technology, vol. 24, 2016, pp. 1499–1506.

[3] B.Samson, E.Ong, "Extracting Conceptual Relations from Children's Stories", in Knowledge Management and Acquisition for Smart Systems and Services. PKAW, vol. 8863, 2014, pp. 195–208.

[4] P. Genest, P. Portier, E. Egyed-Zsigmond, L. Goix, "Promptore: A Novel Approach Towards Fully Unsupervised Relation Extraction", in CIKM '22: The 31st ACMInternational Conference on Information and Knowledge Manage ment, Oct. 2022, pp. 561–571.

[5] C.Zhang,J. Lyu, K. Xu, "A Story Tree-based Model for Inter-document Causal Relation Extraction from News Articles", Knowl Inf Syst, vol. 65, 2023 pp. 827–853.

[6] S.Shahsavari, E. Ebrahimzadeh, B. Shahbazi, B. Falahi, P. Holur, R. Bandari, "An Automated Pipeline for Character and Relationship Extraction from Readers Literary Book Reviews", 2020. Erişim Tarihi: 20 Mayıs 2024. [Çevrimiçi]. Mevcut: https://escholarship.org/uc/item/8b35s9wm.

[7] K. Janakiraman, "Extracting Character Relationships from Stories", 2009. Erişim Tarihi: 21 Mayıs 2024. [Çevrimiçi]. Mevcut: https://api.semanticscholar.org/CorpusID:10981919.

[8] H. Vala, D. Jurgens, A. Piper, D. Ruths, "Mr. Bennet, His Coachman, and The Archbishop Walk into A Bar but only One of Them Gets Recognized: On the Difficulty of Detecting Characters in Literary Texts", in Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, 2015, pp. 769–774.

[9] D. Ong, W. Lui, "Building an Emotional Relation Extraction Tool", 2013. Erişim Tarihi: 21 Mayıs 2024. [Çevrimiçi]. Mevcut: https://nlp.stanford.edu/courses/cs224n/2013/reports/.

[10] D. Christou, G. Tsoumakas, "Extracting Semantic Relationships in Greek Literary Texts", Sustainability, vol. 13, no. 16, 2021.

[11] A. Bajracharya, S. Shrestha, S. Upadhyaya, B. Shrawan, S. Shakya, "Automated Characters Recognition and Family Relationship Extraction from Stories", in 8th International Conference on Cloud Computing, Data Science &Engineering (Confluence), 2018, pp. 14–15.

[12] M. Nijila, M. Kala, "Extraction of Relationship between Characters in Narrative Summaries", in International Conference on Emerging Trends and Innovations in Engineering and Technological Research (ICETIETR), 2018, pp. 1–5.

[13] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is All You Need”, Advances in Neural Information Processing Systems, 2017.

[14] S. Özkaya, B. Diri, “Named Entity Recognition by Conditional Random Fields from Turkish Informal Texts”, in 2011 IEEE 19th Signal Processing and Communications Applications Conference (SIU), Antalya, Turkey, 2011, pp. 662–665. Doi: 10.1109/SIU.2011.5929737.

[15] S. Aslan, “Derin Öğrenme ile Duygu Analizi”, 2020. Erişim Tarihi: 22 Mayıs 2024. [Çevrimiçi]. Mevcut: https://www.veribilimiokulu.com/derin-ogrenme-ile-duygu analizi/.

[16] S. Kim, H. Wimmer, J. Kim, “Analysis of Deep Learning Libraries: Keras, Pytorch, and Mxnet”, in IEEE/ACIS 20th International Conference on Software Engineering Research, Management and Applications (SERA), Las Vegas, NV, USA, 2022, pp. 54–62. Doi: 10.1109/SERA54885.2022. 9806734.

[17] S. Ravichandiran, “Getting Started with Google BERT: Build and train state-of-the-art natural language processing models using BERT”, Packt Publishing, 2021.

[18] Zeyrek, “Zeyrek 0.1.0 documentation”. Erişim Tarihi: 22 Mayıs 2024. [Çevrimiçi]. Mevcut: https://zeyrek.readthedocs.io/en/latest/.

[19] spaCy,” Spacy 3.0 documentation”. Erişim Tarihi: 23 Mayıs 2024. [Çevrimiçi]. Mevcut: https://spacy.io/api/doc.

[20] X. Jin, S. Zhang, J. Liu, “Word Semantic Similarity Calculation Based On word2vec”, in 2018 International Conference on Control, Automation and Information Sciences (ICCAIS), Hangzhou, China, 2018, pp. 12–16. Doi: 10.1109/ICCAIS.2018.8570612.

[21] Tinkerd, “BERT Embeddings”. Erişim Tarihi: 23 Mayıs 2024. [Çevrimiçi]. Mevcut: https://tinkerd.net/blog/machine-learning/bert-embeddings/.

[22] J. Xiao, Z. Zhou, “Research Progress of RNN Language Model”, in 2020 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA), Dalian, China, 2020, pp. 1285–1288. Doi: 10.1109/ICAICA50127.2020.9182390.

[23] A. Mittal, “Understanding RNN and LSTM”, 2019. Erişim Tarihi: 23 Mayıs 2024. [Çevrimiçi]. Mevcut: https://aditi-mittal.medium.com/understanding-rnn-and-lstm-f7cdf6dfc14e.

[24] S. Amidi, “Cs-230 Recurrent Neural Networks Cheatsheet”. Erişim Tarihi: 23 Mayıs 2024. [Çevrimiçi]. Mevcut: https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-recurrent-neural-networks.

[25] A. Zhang, Z. C. Lipton, M. Li, A. J. Smola, “Long short-term memory (LSTM)”. Erişim Tarihi: 23 Mayıs 2024. [Çevrimiçi]. Mevcut: https://d2l.ai/chapter_recurrent-modern/lstm.html.

[26] A. Zhang, Z. C. Lipton, M. Li, A. J. Smola, “Transformer”. Erişim Tarihi: 23 Mayıs 2024. [Çevrimiçi]. Mevcut: https://www.d2l.ai/chapter_attention-mechanisms-and transformers/transformer.html.

A
MASAL METİNLERİ

Örnek masal - Lustig

Bir zamanlar büyük bir savaş vardı ve sona erdiğinde birçok asker terhis edildi. Sonra, **Lustig** de terhis edildi ve yanında sadece küçük bir mermi ekmek ve dört kreuzer para bulunan bir askerlik ekmeği aldı, buyla ayrıldı. Ancak **Aziz Petrus**, ona bir dilenci kılığında yolun üstüne yerleşmişti ve **Lustig** oraya gelince, ondan sadaka istedi. **Lustig**, "Sevgili dilenci, sana ne vereyim? Ben bir askerdim, terhis oldum ve sadece bu küçük mermi ekmeği ve dört kreuzer param var. Bu bittiğinde senin gibi dilenci olmak zorunda kalacağım. Yine de sana bir şey vereceğim." Bunun üzerine ekmeği dört parçaya böldü ve **Aziz Petrus'a** bir tanesini ve bir kreuzeri de verdi. **Aziz Petrus** ona teşekkür etti, devam etti ve bir dilenci olarak tekrar karşısına çıkarak ondan bir hediye istedi. **Lustig** önceki gibi konuştu ve bir dilim ekmek ve bir kreuzer daha verdi. **Aziz Petrus** teşekkür etti ve devam etti, ancak bu sefer bir dilenci olarak başka bir kılıkta **Lustigin** yoluna çıkarak ondan bir hediye istedi. **Lustig**, "Şimdi nereden bulabilirim?" diye cevap verdi. "Terhis oldum ve sadece bir mermi ekmeği ve dört kreuzer param var. Yolda üç dilenciyle karşılaştım ve her birine ekmekten bir dilim ve bir kreuzer verdim. Son dilimi locanda yedim ve son kreuzerimle bir içki içtim. Şimdi cebim boş ve eğer senin de bir şeyin yoksa birlikte dilenmeye gidebiliriz." "Hayır," diye cevap verdi **Aziz Petrus**, "bunu tam anlamamıza gerek yok. Ben biraz tıp hakkında bilgi sahibiyim ve bununla ilgili hızlıca bir şeyler kazanabilirim. Gerçekten, buna ihtiyacımız yok." "Anlaşıldı," dedi **Lustig**, "ve eğer bir şey kazanırsam, sana yarısını vereceğim." "Tamam," dedi **Lustig**, ve birlikte yola çıktılar. Sonra bir köylü evine geldiler, içeride yüksek sesle feryatlar ve çığlıklar duydular. Girdiler ve orada kocası ölüm döşeğinde yatıyordu ve sona yaklaşıyordu, karısı ağlıyor ve çığlık atıyordu. **Aziz Petrus**, "Bu ağlama ve çığlıkları bırakın," dedi. "Adamı anında iyileştireceğim." Yüksek sevinç içinde adam ve karısı, "Sizi nasıl ödüllendirebiliriz? Size ne verebiliriz?" dediler. Ancak **Aziz Petrus** hiçbir şey almak istemedi ve ne kadar teklif ederlerse etsin reddetti. **Lustig**, "Aziz kardeş, bir şeyler almalısınız" diye fısıldadı, "Evet, ihtiyacımız var." Sonunda kadın bir kuzu getirdi ve ona gerçekten bunu alması gerektiğini söyledi, ama o istemedi. **Lustig** bir tokat attı ve "Al işte,

aptal! Gerçekten ihtiyacımız var" dedi. **Aziz Petrus** en sonunda, "Peki, kuzuyu alacağım, ama taşımamı istemiyorum. Eğer senin almak istiyorsan, sen taşı." "Bunu taşımak hiçbir şey," dedi **Lustig**. "Kolayca taşırım." Ve omzuna aldı. Sonra yola çıktılar ve **Lustig** bir ormana geldi, ama kuzu ağır hissetmeye başlamıştı ve açtı, bu yüzden **Aziz Petrus**'a dedi ki, "Bak, işte güzel bir yer, kuzuyu burada pişirebiliriz ve yeriz. Nasıl istersen." "İstersen öyle olsun," diye cevap verdi **Aziz Petrus**, "Ama ben pişirme işiyle ilgilenmiyorum. Sen pişirirsen, senin için bir tencere var ve ben hazır olana kadar biraz dolaşayım." Ama yemeye başlamadan önce geri gelmeden önce beklememem için uyarıda bulundu. "Tamam, git o zaman," dedi **Lustig**. "Ben yemek yapmayı bilirim, bununla başa çıkarım." Sonra **Aziz Petrus** uzaklaştı ve **Lustig** kuzunun etini çıkardı, ateşe attı ve kaynattı. Ancak kuzu tamamen hazır olduğunda ve **Aziz Petrus** geri gelmemişse, **Lustig** tencereden çıkardı, doğradı ve kalbine ulaştı. "Bu dedikleri gibi en iyi kısım olmalı," dedi ve onu tadarak yedi, ancak sonunda hepsini yedi. Sonunda **Aziz Petrus** geri döndü ve dedi ki, "Tüm kuzu senin, yalnızca kalbi bana ver." Sonra **Lustig** bıçak ve çatal aldı ve etin içinde gerçekten bulmak için baktığına dair endişeli bir ifade takındı, ancak onu bulamamış gibi göründü ve sonunda birdenbire, "Burada yok" dedi. "Ama nerede olabilir ki?" dedi **Aziz Petrus**. "Bilmiyorum," diye cevap verdi **Lustig**, "ama kalbi kayboldu, bunu biliyorum." "O zaman kaybettiğin şeyi bulmana yardım edeyim," dedi **Aziz Petrus**, ve **Lustig** ne demek istediğini anlamadan bir hokus pokus yaptı ve birdenbire masanın üzerinde kalp belirdi. "Haydi, bu senin," dedi Aziz Petrus, "ve gerçekten bana ihtiyacın olacak gibi görünüyor." Sonra kayboldu ve bir daha hiç görülmedi. **Lustig** geri döndü, kuzunun etini yedi ve eve döndü. Ancak **Lustig**, **Aziz** Petrus'un ona olan iyiliğini unutmadı. Artık iyi bir insan olmaya ve herkese yardım etmeye karar verdi. O günden sonra, dilencilere ve yoksullara yardım etti, aç olanlara yemek verdi ve iyilik yaparak ömrünü geçirdi. **Aziz Petrus**'un lütfu sayesinde **Lustig** gerçek bir kahraman oldu ve insanların kalplerini kazandı.

Diğer masallara Google Drive bağlantısı kullanılarak erişilebilir:

“https://drive.google.com/drive/folders/1XS4ZJu5OLapQixWRtjU95qNzhZviGkhE?usp=drive_link”

Elle Etiketleme	**Geliştirilen Sistemin Buldukları**
Birinci Kahraman: Lustig	Birinci Kahraman: Lustig
İkinci Kahraman: Petrus	İkinci Kahraman: Petrus
Diğer Kahramanlar: Köylü	Diğer Kahramanlar: Bulunamamıştır
Duygusal İlişki	**Duygusal İlişki**
Lustig, Aziz Petrus: pozitif	Lustig, Aziz Petrus: pozitif
Lustig, Köylü: pozitif	Bulunamamıştır
Petrus, Köylü: pozitif	Bulunamamıştır

TEZDEN ÜRETİLMİŞ YAYINLAR

Konferans Bildirileri

1. S. Ganbarli and B. Diri, "Sentiment Relation Detection between Main Characters in Turkish Stories," 2024 32nd Signal Processing and Communications Applications Conference (SIU), Mersin, Turkiye, 2024, pp. 1-4, Doi: 10.1109/SIU61531.2024.10600820.