FEN BİLİMLERİ ENSTİTÜSÜ

NLP KULLANILARAK HABERLERİN YAŞ GRUPLARINA GÖRE SINIFLANDIRILMASI

Rabia KONTUK

Danışman
Dr. Öğr. Üyesi Metin TURAN

YÜKSEK LİSANS TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
İSTANBUL - 2020

KABUL VE ONAY SAYFASI

Rabia KONTUK tarafından hazırlanan **"NLP Kullanılarak Haberlerin Yaş Gruplarına Göre Sınıflandırılması "** adlı tez çalışması 13/07/2020 tarihinde aşağıdaki jüri üyeleri önünde başarı ile savunularak, İstanbul Ticaret Üniversitesi Fen Bilimleri Enstitüsü **Bilgisayar Mühendisliği Anabilim Dalı**'nda **Yüksek Lisans Tezi** olarak kabul edilmiştir.

Danışman	**Dr. Öğr. Üyesi Metin TURAN** İstanbul Ticaret Üniversitesi
Jüri Üyesi	**Dr. Öğr. Üyesi Arzu KAKIŞIM** İstanbul Ticaret Üniversitesi
Jüri Üyesi	**Prof. Dr. Selim AKYOKUŞ** Medipol Üniversitesi

Onay Tarihi : 24/07/2020

İstanbul Ticaret Üniversitesi, Fen Bilimleri Enstitüsünün 24.07.2020 tarih ve 2020/288 numaralı Yönetim Kurulu Kararının 1. maddesi gereğince, ders yüklerini ve tez yükümlülüğünü yerine getirdiği belirlenen Rabia Kontuk (TC: 44647414004) adlı öğrencinin mezun olmasına oy birliği ile karar verilmiştir.

Prof. Dr. Necip ŞİMŞEK
Enstitü Müdürü

AKADEMİK VE ETİK KURALLARA UYGUNLUK BEYANI

İstanbul Ticaret Üniversitesi, Fen Bilimleri Enstitüsü, tez yazım kurallarına uygun olarak hazırladığım bu tez çalışmasında,

- tez içindeki bütün bilgi ve belgeleri akademik kurallar çerçevesinde elde ettiğimi,
- görsel, işitsel ve yazılı tüm bilgi ve sonuçları bilimsel ahlak kurallarına uygun olarak sunduğumu,
- başkalarının eserlerinden yararlanılması durumunda ilgili eserlere bilimsel normlara uygun olarak atıfta bulunduğumu,
- atıfta bulunduğum eserlerin tümünü kaynak olarak gösterdiğimi,
- kullanılan verilerde herhangi bir tahrifat yapmadığımı,
- ve bu tezin herhangi bir bölümünü bu üniversitede veya başka bir üniversitede başka bir tez çalışması olarak sunmadığımı

beyan ederim.

24/07/2020

Rabia KONTUK

İÇİNDEKİLER

ÖZET

Yüksek Lisans Tezi

NLP KULLANILARAK HABERLERİN YAŞ GRUPLARINA GÖRE SINIFLANDIRILMASI

Rabia KONTUK

İstanbul Ticaret Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Dr. Öğr. Üyesi Metin TURAN

2020, 74 sayfa

Çalışma kapsamında haber metinlerinin yaş gruplarına göre Doğal Dil İşleme tekniğinden faydalanılarak sınıflandırılması sağlanmıştır. Çünkü gelişen teknoloji ile beraber özellikle çocuk yaş grubunun, zarar görebileceği içeriklerden uzak tutulması gerekmektedir. Her ne kadar İnternet Servis Sağlayıcıları ailelere filtreleme imkanları sunsa da ailelerin çoğu bu filtrelemeyi uygulamakta zorlanmakta veya kayıtsız kalmaktadırlar. Bu tür olumsuz durumların üstesinden gelmek için internette yayınlanan içerikler üzerinde yasal bir kontrol sistemi gerekmektedir. Python dili kullanılarak geliştirilen çalışmada Türkçe haber metinlerinin Doğal Dil işlemleri için Zemberek Kütüphanesi kullanılmıştır. Havighurst'ün Gelişim Kuramından faydalanılarak Çocukluk, Ergenlik ve Yetişkinlik yaş grupları belirlenmiştir. Belirlenen yaş gruplarına ait haberlerin bulunduğu toplamda 3925 haber öğesini içeren bir veri kümesi oluşturulmuştur. Veri kümesinin eğitim haberleri ile sözlük oluşturulup sınama haberleri ile de sözlük test edilmiştir. İlk test işleminde haberin yaş grubunu belirleme doğruluğu %71 olarak bulunmuştur. Gözlem doğrultusunda, sadece isimleri içeren sözlük ile %73'lük bir başarı elde edilmiştir. Diğer bir gözlem doğrultusunda, Ergenlik yaş grubuna ait kelimelerin diğer iki grupla örtüşmesi nedeniyle, Ergenlik yaş grubu ile Çocukluk yaş grubu birleştirilip yetişkin ve yetişkin olmayan yeni yaş grupları oluşturulmuş, sadece isimleri barındıran bir sözlük ile %83 oranında daha ayrıştırıcı bir sonuç elde edilmiştir.

Anahtar Kelimeler: Doğal Dil İşleme, haber yaş grubu tespiti, terim frekansı, yaş grubu sözlüğü, Zemberek.

ABSTRACT

M.Sc. Thesis

CLASSIFICATION OF NEWS ACCORDING TO AGE GROUPS USING NLP

Rabia KONTUK

İstanbul Commerce University
Graduate School of Applied and Natural Sciences
Department of Computer Engineering

Supervisor: Assoc. Prof. Dr. Metin TURAN

2020, 74 pages

Within the scope of the study, it was provided to classify news texts according to age groups by using Natural Language Processing technique. Because with the developing technology, especially the child age group should be kept away from the content that can be damaged. Although Internet Service Providers provide families with filtering facilities, most families find it difficult to implement this filter or remain indifferent. In order to overcome such negative situations, a legal control system is required on the content published on the internet. In the study developed using Python language, Zemberek Library was used for Natural Language operations of Turkish news texts. Childhood, Adolescence and Adulthood age groups were determined by using Havighurst's Development Theory. A dataset containing 3925 news items was created in which there are news belonging to the determined age groups. The educational news of the dataset was created and a dictionary was tested with the test news. In the first test process, the accuracy of determining the age group of the news was found to be 71%. In line with the observation, a 73% success was achieved with a dictionary containing only names. In line with another observation, since the words belonging to the Adolescence age group overlap with the other two groups, the Adolescence age group and the Childhood age group have been combined to create new adult and non-adult age groups, resulting in a distinctive result of 83% with a dictionary containing only names.

Keywords: Age group dictionary, Natural Language Processing, news age group detection, term frequency, Zemberek.

TEŞEKKÜR

Bu araştırma için beni yönlendiren, karşılaştığım zorlukları bilgi ve tecrübesi ile aşmamda yardımcı olan değerli zamanını ayırıp tez sürecimde yaşadığım tüm eksikliklere çözüm bulan kıymetli Danışman Hocam Dr. Öğr. Üyesi Metin TURAN'a teşekkürlerimi sunarım.

Tezimin her aşamasında beni yalnız bırakmayan aileme ve arkadaşlarıma sonsuz sevgi ve saygılarımı sunarım.

Rabia KONTUK
İSTANBUL, 2020

ŞEKİLLER

ÇİZELGELER

SİMGELER VE KISALTMALAR

API	Application Programming Interface
BTK	Simge veya Kısaltma açıklaması
DDİ	Doğal Dil İşleme
HTML	Hyper Text Markup Language
IDE	Integrated Development Environment
ISS	İnternet Servis Sağlayıcı
JSON	JavaScript Object Notation
MPL	Mozilla Kamu Lisansına
NLP	Natural Language Processing
PDR	Psikolojik Danışma ve Rehberlik
POS	Part of Speech
TF	Terim Frekansı
XML	Extensible Markup Language

1. GİRİŞ

İnternet ağıyla günümüzde istediğimiz her türlü bilgiye kolay bir şekilde ulaşmak mümkündür. Ancak, internet üzerinde bulunan bu bilgi havuzunda doğru/yanlış veya güncel/eski ayrımı yapılmaksızın her türlü içerik bulunması, istenen bilgiye ulaşmayı zorlaştırmaktadır. Bu durum, interneti kullanan kitleler için bilgi kirliliğinin içinde doğru bilgiye ulaşma ihtiyacı ve arayışı ortaya çıkarmıştır. Daha da önemlisi, sanal dünyadaki bu kuralsız ve sınırsız paylaşımların internet ağında var olan her kitleye bir tık uzaklıkta olması, aslında bu içerikler için bir düzenleme, hatta bir sınırlandırma olması gerekliliğini ortaya koymaktadır.

Office of Communications veya bilinen kısa adıyla Ofcom'un, çocukların medya okuryazarlığına dair yaptığı 2019 yılına ait çalışmada internete erişim sağlayan çocukların yaş gruplarına dair bilgiler vermektedir (Ofcom, 2019). Çalışmada, 3-4 yaş grubundaki çocukların %53'ü, 5-7 yaş grubundaki çocukların % 79'u, 8-11 yaş grubundaki çocukların % 94'ü ve 12-15 yaş grubundaki çocukların %99'unun çevrimiçi ortamlarda yer aldığından bahsedilmektedir. Bu oranların gittikçe büyümesinin nedeni olarak, teknolojinin hayatımıza kazandırdığı tabletler belirtilmiştir. Bu çalışma, bizlere internette olan kitlenin 3 yaşına kadar düştüğünü ve aslında internet üzerinde var olan her türlü içeriğin kullanıcıya (özellikle yetişkin olmayan bireylerin) çok dikkatli bir şekilde sunulmasının gerekliliğini göstermekte, tezin ana fikri olan içeriklerin sınırlandırılma fikrini doğrular niteliktedir.

Bu noktada akla ilk gelen yöntem, kendimizi ve yakınlarımızı korumak üzere İnternet Servis Sağlayıcılar (ISS) tarafından veya bizzat bizim uyguladığımız kişisel filtreler olsa da, bunlar genellikle cinsel içerikli kaynaklar üzerinde yoğunlaşmıştır. Bir diğer ayrıntı, Demirel vd. (2013) tarafından yapılan çalışmaya göre, deney grubunun %81,4'ü Bilgi Teknolojileri ve İletişim Kurumu'nun (BTK) Güvenli İnternet hizmetini desteklemektedir. Bunların sadece %36,9'u gibi çok az bir oran filtre kullanmaktadır. Sonuç olarak deney

grubunun çoğunluğu BTK'nın internet filtresi hizmetini desteklemekte, ancak bu hizmetten yararlanma konusunda çok da istekli olmadığı görülmektedir.

İnternetin günümüzde bu denli yaygın kullanımı bizlere beraberinde birçok yeni alışkanlıklar da kazandırmıştır. Buna en güzel örnek, yazılı basın organlarının yerini artık elektronik olan haber sitelerine (e-haberlere) bırakmış olmasıdır. Aslında dijitalleşmeyle birlikte kalemlerin yerini alan klavyeler, kağıtların yerini alan ekranlar, e-haber kavramını insanlığa hazırlamış oldu. Açıkçası bu durum, günlük yaşam içinde yer alan ve hiç de ahlaki olmayan veya şiddet içeren haberleri de kontrol altına alabilmenin ne kadar önemli olduğunu göstermektedir; Çünkü ne pahasına olursa olsun ama okunsun yaklaşımı yayıncılık kalitesini düşürmektedir (Işık ve Koz, 2013).

Elektronik haberler için internetin haber üretimi ve sunumunu kolaylaştırıcı olanakları sayesinde, basılı gazetelere kıyasla çok daha fazla habere yer verilebilmektedir. Bununla birlikte, hem habercilerin daha fazla ve anlık haber sunma kaygısı, hem de yeterli mesleki bilgiye sahip olmaması ve daha da önemlisi tıklanma sayısını artırma kaygısı haberin niteliksel ve etik unsurlarının ihmal edilmesine sebep olabilmektedir. Bu hatalar, söz konusu kullanıcının yetişkin olmadığı düşünüldüğünde çok daha önemli olmaktadır (Fırat, 2016).

Haberlerin niteliksel ve etik unsurları ihmal etmesi göz önüne alınarak, tezde doğal dil işleme ile elektronik haberlerin yaş gruplarına göre sınıflandırılması (kullanıcı yaşına uygun haber gösterimi) problemi üzerinde çalışılmıştır.

Yaş grupları belirlenirken çalışmada gönüllü olarak yer alan Psikolojik Danışman ve Rehberlik (PDR) uzmanı tarafından çalışma için uygun görülen Havighurst'ün "Gelişim Kuramı" temel alınmıştır (Çizelge 1.1). Havighurst'ün Gelişim Kuramında 6 kategoriye ayrılan yaş grupları, okuma yazma yaşı göz önünde bulundurularak (ilkokul çağı başlangıcı) yeniden düzenlenip çocukluk (6-13 yaş), ergenlik (13-18) ve yetişkinlik (18+) olmak üzere 3 kategoriye indirgenmiştir.

Çizelge 1.1. Havighurst'ün gelişim kuramı (Çok, 1993)

Yaklaşık Yaş	Karakteristik Görevler
Erken Çocukluk (Doğum - 6 yaş)	Yürümeyi, katı yiyecek yemeyi, konuşmayı, tuvalet kullanmayı öğrenir; cinsiyet farklarının toplumsal doğrularını öğrenir. Bu evrenin sonuna doğru, daha kavramsal görevler beklenir, doğruyu yanlıştan ayırabilir, vicdan geliştirmeye başlar, okumayı öğrenmeye hazırlanır; işaretlerin (örneğin gülümseme) kelimelerin yerine geçebildiğini öğrenir.
Orta çocukluk (6-12/13 yaşlar)	Oyun oynamada gerekli fiziksel becerileri öğrenir, akranlarla birlikte olmayı öğrenir; uygun kadınsı ya da erkeksi toplumsal rolü öğrenir, okuma, yazma ve hesaplamada temel becerileri geliştirir; vicdan, ahlak ve değerler geliştirmeye devam eder, özerklik geliştirmeye başlar, temelde demokratik olan toplumsal tutumlar geliştirir.
Ergenlik (13-18 yaşlar)	Duygusal ve fiziksel olarak olgunlaşır. Her iki cinsten akranlarla olgun ilişkiler kurar, toplumsal olarak kabul edilen erkeksi ya da kadınsı toplumsal rolü öğrenir. Fiziğini kabul eder ve bedenini etkili biçimde kullanır, ana babadan ve diğer yetişkinlerden duygusal bağımsızlık kazanır, evlilik ve aile yaşamı için hazırlanır, bir mesleğe hazırlanır ve davranışlarını belirleyecek bir dizi değer ve ilke kazanır.
Erken Yetişkinlik (18-35 yaşlar)	Eş seçer, yakın bir ortakla yaşamayı öğrenir, yuva kurar, mesleğine başlar, vatandaşlık sorumluluklarını kabul eder, toplumsal ilişkiler kurar.
Orta Yaş (35-60 yaşlar)	Ergenlerin sorumlu kişiler olmasına yardım eder; toplumsal ve vatandaşlık sorumluluğu kazanır, mesleki doyum elde eder, orta yaşın fiziksel değişikliklerine uyum sağlar, yaşlanan ana babaya uyum sağlar.
İleri Olgunluk (60 yaşın ötesi)	Azalan fiziksel güce, eşin ölümüne, azalan ya da sabit kalan gelire uyum sağlar, akranlarıyla yakınlık kurar.

1.1. Çalışmanın Amacı

Tez motivasyonunu, haber sitelerinin anlık haber yayınlama kaygısı ile haberlerin niteliksel ve etik unsurlarını (cinsellik, küfür, argo, tecavüz, silah, şiddet vb.) ihmal etmesi ve yayınlanan haberlerin her yaş grubu tarafından okunmasının sakıncası oluşturmaktadır. Özellikle çocukların, hem ahlaki hem de psikolojik anlamda yayınlanan içeriklerden zarar görmemesi, örnek teşkil edip özendirici olmaması adına, yaş gruplarına uygun haberlerin okunabilir olması gereklidir. Bildiğimiz kadarı ile dünyada bu amaçla uygulanan ilk çalışma olarak da önemlidir. Tezde mevcut probleme çözüm olmak üzere, Türkçe

haberlerin ilgili yaş gruplarına uygun olarak sınıflandırılmasına yönelik bir model önerilmiş ve uygulanmıştır.

2. LİTERATÜR ÖZETİ

Çözüm önerisinde, hem psikoloji alanında bireylerin yaş gruplarının dönemlere ayrıldığı çalışmalar, hem de Doğal Dil İşleme alanındaki sözlük çalışmaları dikkate alınmıştır. Göz önüne alınacak yaş gruplarının belirlenmesi için gönüllü Psikolojik Danışma ve Rehberlik (PDR) uzmanı [1] ile Havighurt Gelişim Kuramının dışında farklı kuramlar da incelenmiştir.

Bilişsel gelişim üzerine ciddi çalışmalar yapan psikologların başında gelen Piaget kuramında, bilişsel gelişimi dört döneme ayırmıştır. Bunlar; Duyusal Motor Dönemi (0-2 yaş), İşlem Öncesi Dönem (2-7 yaş), Somut işlemler Dönemi (7- 11 yaş) ve Soyut İşlemler Dönemi (12 yaş ve üzeri) şeklinde sıralanır (Kol, 2011).

Erikson insanın psikososyal evreler içinde gelişimini devam ettirdiğini ileri sürmektedir. Bu amaçla Erikson Psiko-Sosyal Gelişim Kuramı'nı hazırlamıştır. Bu kuramda 8 aşama bulunmaktadır. Bunlar; Temel Güvene Karşı Güvensizlik Duygusu (0-1 yaş), Özerkliğe Karşı Kuşku ve Utanç Duygusu (1-3 yaş), Girişimciliğe Karşı Suçluluk Duygusu (3-6 yaş), Başarılı Olmaya Karşı Yetersizlik Duygusu (7-11 yaş), Kimlik Kazanmaya Karşı Kimlik Karmaşası (11-17 yaş), Yakınlığa Karşı Yalıtılmışlık (17-30 yaş), Üretkenliğe Karşı Durgunluk (30-60 yaş) ve Benlik Bütünlüğüne Karşı Umutsuzluk (60 yaş ve üzeri) olarak sıralanır (Gürses ve Kılavuz, 2011).

Psikoloji alanındaki en önemli isimlerin başında gelen Freud, Psikoseksüel Kuramı'nı ortaya atmıştır. Freud'a göre yeni doğan bir bireyin kişiliği farklı aşamalardan geçerek gelişmektedir. Bu aşamaları Oral Dönem (0-1), Anal Dönem (1-3 yaş), Fallik Dönem (3-6 yaş), Latens (gizlilik) Dönemi (6-11 yaş) ve Genital (ergenlik) Dönem(11 yaş ve üzeri) olmak üzere beş gruba ayırmıştır (Özdemir vd., 2012).

[1] Asu Akdemir, MBA Okulları

Bir diğer önemli psikolog olan Bruner bilişsel gelişim üzerine çalışmalar yapmış ve bilişsel gelişimi üç döneme ayırmıştır. Bruner'in Bilişsel gelişim kuramında dönemler Eylemsel Dönem (0-3 yaş), İmgesel Dönem (4-6 yaş) ve Sembolik Dönem (7 yaş ve üzeri) şeklinde sıralanmaktadır (Çekirdekci vd., 2016).

Doğal Dil İşleme alanında, sözlük oluşturma üzerine eskilerden günümüze kadar çeşitli çalışmalar yapılmıştır. Bu çalışmalar farklı diller için uygulansa da, çoğunlukla İngilizce dili üzerine yapılmıştır. Literatürde var olan sözlük çalışmalarını, Türkçe ve diğer diller olmak üzere iki grup olarak incelemek daha faydalı olacaktır.

İlk çalışma Doğal Dil İşleme sözlük çalışmalarının temeli, Miller vd. (1985) tarafından Princeton Üniversitesi Bilişsel Bilimler Laboratuvarı'nda hazırlanan "Wordnet" adlı sözlüktür. Öncelikle İngilizce dili için oluşturulan bu sözlük, daha sonra farklı birçok dil seçeneğini de barındırmaktadır. Ücretsiz, kolay erişilebilirliği ve indirilebilir olması bu sözlüksel veri tabanını günümüzde birçok çalışmanın ilk durağı haline gelmiştir (WordNet, 2020).

Literatürde "Levinin Sınıfları" olarak geçen ve sözlük çalışmalarının çok önemli kaynaklarından biri haline gelen çalışmada Beth Levin, 1993 yılında İngilizce dilinde bulunan 3.000 adet fiili anlam ve davranışa göre sınıflandırmıştır. Beth Levin, çalışmaya bir fiilin anlamının sözdizimsel davranışını etkilediği fikriyle başlar ve onu fiil sözlüğü üzerinde çalışmak için güçlü bir araca dönüştürür. Birbirine benzer sözdizimsel davranışa sahip fiillerin tanımlanmasının, anlamsal olarak tutarlı fiil sınıflarını ayırt etmek için etkili bir yol sağladığını gösterir. Ayrıca fiillerin anlamını yansıtan çok çeşitli sözdizimsel değişimlere göre fiillerin davranışını inceleyerek bu sınıfları yalıtır (The University of Chicago Press, 2020).

Literatürün bir diğer önemli kaynaklarından biri de FrameNet'dir. FrameNet, Kaliforniya Berkeley'deki Uluslararası Bilgisayar Bilimi Enstitüsü'nde yer alan ve çerçeve anlambilimi adı verilen anlam teorisine dayanan, elektronik kaynak üreten büyük bir projedir. Online erişim imkanı da sağlanan sözlüğün

(FrameNet, 2020) proje liderliğini Charles J. Fillmore üstlenmiştir. FrameNet sözcük veri tabanı, 1.200'den fazla semantik çerçeve, 13.000 sözcük birimi ve 202.000 cümle içermektedir. FrameNet, örneğin, "Adam'ın July'e bir araba sattığı" cümlesinin, temelde "July, Adam'dan bir araba aldı" ile aynı temel durum (anlamsal çerçeve) olduğunu ortaya koymaktadır (Wikipedia, 2020).

Silverman vd. (1999) tarafından ABD İngilizcesi için geliştirilen "Victoria" adlı sözlüğün tasarımı ve koleksiyonu tanıtılmıştır. Bu sözlük, Apple Computer'da konuşma sentezi araştırma ve geliştirmesini desteklemek amacı ile oluşturulmuştur ve MacinTalk 4 için adım ve süre modellerinin tahmininde kullanılmaktadır. Çalışmada üretilen sonuçlarda ise, böyle bir sözlüğün oluşturulmasında gerçek dünya koşullarının çoğunu karakterize eden gürültü vb. etkenlerden dolayı daha düşük sentez kalitesine neden olacağı ve böyle bir konuşma sentezi sözlüğünün yaratılmasında çok büyük zorluklar oluşturacağı bildirilmektedir.

1999 yılındaki bir diğer çalışma ise İngilizce dili için hazırlanmış "AutoSlog" adlı bir sözlüktür (Riloff, 1999). Sözlüğün amacı verilen metinlerden bilgi çıkarımı yapmaktır. Bu amaç doğrultusunda AutoSlog sözlüğü ile terörizm metinleri ve cevap anahtarlarının (MUC-4) üzerinden bilgi çıkarımı yapılmıştır. AutoSlog sözlüğü iki üniversite öğrencisinin hazırladığı el yapımı başka bir sözlük ile karşılaştırıldığında %90'ın üzerinde bir sonuca ulaşmıştır.

Hanks Patrick ve Pustejovsky James kelime algılaması için var olan kaynaklara eleştiri yaparak yeni bir yaklaşım önermişlerdir (Hanks ve Pustejovsky, 2005). Bu yeni yaklaşım ile İngilizce dili için bir fiilin tüm olası kullanımları değil, tüm normal kullanımları açıklanmaktadır. Corpus Pattern Analysis (CPA) ile fiillerin normal kullanım desenleri belirlenir ve her desenle bir anlam ilişkilendirilir. Desenler daha sonra, herhangi bir cümlenin olası anlamının ölçülebileceği kıyas ölçütleri olarak kullanılır.

2011 yılındaki bir diğer çalışma ile İngilizce dili için bir konuşma grubu oluşturulmuştur (Këpuska ve Rojanasthien, 2011). Bu çalışmada, film ve

dizilerin Digital Versatile Disc (DVD)'leri kullanılarak konuşma grubu oluşturmak için bir sistem sunulmuştur. Bu sistem konuşma tanıma, tonlama, vurgu, perde, duraksama yapabilmektedir. Çalışmanın avantajı ise DVD'ler kullanılarak bir sözlük üretmek ve geleneksel bir konuşma grubu ile sözlük üretmekten çok daha düşük maliyetli olmasıdır. Ek olarak, verilerin toplanmasının ve bir sözlüğe dönüştürülmesinin de çok daha kısa süre alacağı belirtilmiştir. Çalışmanın sonucunda bu sözlük ile bir konuşma grubu oluşturmanın yararlı olduğu gösterilmiştir.

Tsalidis vd. (2014) tarafından Modern Yunanca dili için bir sözlük oluşturulmak istenmiştir. Çalışmanın yapılma amacı, Modern Yunanca'nın morfolojik ve sözdizimsel olarak işlenmesi için elektronik formda bir sözlüğünün olma zorunluluğu hissedilmesidir. Çünkü Modern Yunanca çok yönlü bir dil olmakla beraber eski Yunanca'dan barındırdığı çok fazla karakteristik özelliği bulunan bir dil sistemidir. Çalışma ile Modern Yunanca'yı karakterize eden ve elektronik formdaki Doğal Dil İşlemede (DDİ) geçerli kılan noktaları belirtilmiştir.

2016 yılında Bollegala ve arkadaşları İngilizcede doküman kümesinden (corpus) öğrenilen kelime temsillerini geliştirmek için, anlamsal sözlükte var olan bilgileri kullanmak üzere bir yöntem (sözlükteki anlambilimsel ilişkileri düzenleyici olarak kullanarak küresel bir kelime oluşum tahmin yöntemi) önerdiler (Bollegala vd., 2016). Doküman kümesi olarak ukWaC ve semantik sözlük olarak WordNet kullanılan bu çalışma, yalnızca doküman kümesi kullanılarak öğrenilen kelime temsillerinin önemli ölçüde geliştirebileceği sonucunu ortaya koymuştur.

Chiavetta vd. (2016) İtalyanca dili için Amazon sitesinde bulunan İtalyanca kitapların otomatik olarak duygu analizi (olumlu, olumsuz) sınıflandırılmasını amaçlayan bir sistem sundular. Geliştirilen sistem, sözlük tabanlı bir yaklaşım kullanır ve NLP teknikleri ile kitap içeriklerinin terimler arasındaki dilsel ilişkiyi inceler. İtalyanca için uyarlanmış sözlük eksikliği problemini çözmek için bu çalışmanın yapılmasına ihtiyaç duyulmuştur. Sistem Amazon web-sitesinden

alınan bir veri kümesi üzerinde test edilip % 82'nin üzerinde bir doğruluğa ulaşılmıştır. 2016 yılındaki bir diğer sözlük çalışması ise Svetla Koeva ve arkadaşları tarafından yapılmıştır. Koeva vd. (2016), Bulgarca dilindeki çok kelimeli ifadelerin yarı otomatik olarak bir sözlüğünü oluşturdular. Bu sözlük, çok kelimeli ifadelerin temel özelliklerini, tanımını ve yapısal ve anlambilimsel ölçütlere göre sınıflandırılmasını içermektedir.

Vijay vd. (2018) İngilizce ve Hintçe dili için çevrimiçi olarak yayınlanan tweetleri kullanarak karışık bir sözlük oluşturdular. Bu sözlüğü oluşturmak için, Twitter Python API1 kullanılarak Twitter üzerinden tweetlere eriştiler. Erişilen tweetler JavaScript Object Notation (JSON) formatına dönüştürüldü. Gerekli adımlar tamamlandıktan sonra 2866 kelime içeren sözlük oluşturuldu ve sözlük kelimeleri, belirlenen duygu durumları ile etiketlendi. Çalışma sonunda diğer tweetler için duygu analizi yapılarak %58 oranında bir sonuç kaydedilmiştir.

2018 yılındaki bir diğer çalışma ise Bertin ve Atanassova tarafından hazırlanan InTeReC (In-text Reference Corpus) adını verdikleri İngilizce bir sözlüktür (Bertin ve Atanassova, 2018). Bu sözlük ile referansların farklı kullanımlarını ve var olan makalelerin yapısı ile ilişkilerini gözlemlemek amaçlanmıştır. Amaç doğrultusunda 90.071 makaleden sadece araştırma makaleleri seçilerek 85.660 tane makale sözlük için kullanılmıştır. Çalışmanın cümle düzeyinde sınırlandırılması ve bu nedenle cümle sınırlarını aşan ilgili referans bağlamlarının dikkate alınamadığı belirtilmiştir.

Türkçe, geniş bir coğrafyada çok fazla kişi tarafından anadili olarak konuşulan bir dil olsa da diğer dillerin aksine Türkçe dili için yapılan Doğal Dil İşleme (DDİ) çalışmaları son 15-20 yıl içinde hızlanmıştır (Oflazer, 2012). Türkçe dili üzerine yapılan DDİ sözlük çalışmalarına kronolojik sıra ile yer verilmiştir.

2007 yılında Akın kardeşler tarafından “Zemberek” adlı bir kütüphane oluşturulmuştur (Akın ve Akın, 2007). Bu kütüphane özellikle Türkçe dili için birçok doğal dil işleme çalışmasının temelini oluşturmuştur. Zemberek kütüphanesi yardımı ile metinlerde kelimelere ayrıştırma (Tokenization), dil

bilimi işlemleri (Morphology), durak kelimelerin kaldırılması (Stop Words), kök bulma (Stemming) işlemleri yapılabilmektedir. Bu çalışmada oluşturulan sözlük için Zemberek kütüphanesinden faydalanılmıştır. Aktaş vd. çalışmalarında da yardımcı kütüphane olarak kullanılan Zemberek kütüphanesine çalışmanın ileri ki bölümlerinde detaylandırılarak yer verilecektir.

2014 yılında Gülşen Eryiğit tarafından geliştirilen İTÜ Türkçe Doğal Dil İşleme Yazılım Zinciri, açık kaynak kodlu olmayan Natural Language Processing (NLP) yazılımıdır (Eryiğit, 2014). Sağladığı avantajlar; Türkçe karakter dönüştürücü, sözcük ayrıştırıcı/cümle bölücü, biçimbilimsel çözümleyici/belirsizlik giderici, yazım denetleyici, varlık ismi tanıma ve bağımlılık çözümlemesi vb. şeklinde sıralanabilir. Bu platform, web arayüzüne sahip olmakla beraber Application Programming Interface (API)'e de sahiptir. Böylece farklı seviyelerdeki kullanıcılar bu platformdan faydalanabilir.

2015 yılında Gökhan Çelikkaya tarafından Türkçe doğal dil girdisi ile çalışacak yüksek başarımlı bir mobil asistan uygulaması geliştirilmiştir (Çelikkaya, 2015). Geliştirilen sistem için oluşturulan sözlük temel ve üretken olmak üzere iki yapıdan oluşur. Temel sözlük adı, soyadı, yer, zaman, para birimi gibi bilgileri barındırır. Üretken sözlük unvan, yer, kurum gibi bilgileri barındırır. Model, test verisi üzerinde %98.30 başarı kaydetmiştir.

2015 yılındaki bir diğer çalışma, Akgül ve arkadaşları tarafından Türkçe dili için Twitter verisini ayrıştıran, analiz eden ve raporlayan "Duygusal Twitter" adını verdikleri bir sistemdir (Akgül vd., 2015). Bu sistemde 6800 tweet için olumlu, nötür ve olumsuz olmak üzere üç sonuçtan uygun olan bir sonuç üretilmektedir. Bu çalışmada hem n-gram hem de sözlük kullanılarak iki yöntem geliştirilmiştir ve sözlük yöntemi daha başarılı sonuçlar göstermiştir.

Aktaş, vd. (2017) "WordNet" sözlüğünden de faydalanılarak bir bilişim sözlüğü oluşturmuştur. Bu sözlük ile bilgisayar ağ terimlerinin ontolojik tabanlı oluşturulması işleminin otomatikleştirmesi, Türkçe dilinde sözcükler arası eş anlam, yakın anlam vb. gibi anlamsal bağlantılara sahip sözlüklerin uygun bir

şekilde bir araya getirilmesi üzerine bir çalışma yapılmıştır. Sadece bilgisayar ağ terimleri kullanılan çalışma ile en geniş Türkçe bilişim sözlüğü oluşturulmuştur.

Durna (2019) Türkçe dili için Türkçe haber metinleri kullanarak olay göstergesi tespiti ve türü sınıflandırması yapan çalışmasını tanıtmıştır. Veri seti için çeşitli Türkçe haber siteleri kullanılmıştır. Haberlerdeki her kelime elle dizi türü, gösterge türü, gösterge alt türü, realis değeri (olayın gerçekte olup olmaması) ve ana olay olup olmamasına göre etiketlenip bir veri seti elde edilmiştir. Etiketlenmiş veriler üzerinde sınıflandırma metotları denenmiştir. Türkçe diline özgü morfolojik ve bağlılık ayrıştırma özelliklerinden yararlanılmıştır ve dile özgü özelliklerin sınıflandırmada nasıl bir etkisi olduğu gözlemlenmiştir.

3. YAPAY ZEKA (ARTIFICIAL INTELLIGENCE)

1950 yılında Alan Mathison Turing, Mind adlı bir dergide "Computing Machinery and Intelligence" adlı bir makale yayınlamıştır. Bu makalede İngiliz matematikçisi Alan Mathison Turing "Makineler düşünebilir mi?" sorusunu tartışmaya açarak aslında Yapay Zeka'nın (Artificial İntelligence) temellerini hazırlamış oldu (Pirim, 2006).

Değirmenci'nin kitabında Yapay Zeka tanımı olarak şu ifade yer almaktadır: "Yapay Zeka, canlı bir organizmadan yararlanmadan, tümüyle yapay araçlar ile oluşturulmakta ve insana özgü davranışlar ve hareketler göstererek makinelerin çalışma sistemiyle çalışan teknolojik bir özelliktir. Yapay Zeka; insansı davranışları gösterme, sayısal olarak mantık sağlama, hareket, konuşma ve ses algılama gibi pek çok yeteneği beraberinde taşır. Bu sayede yazılım ve donanım sistemlerini bünyesinde barındırır" (Sucu ve Ataman, 2020).

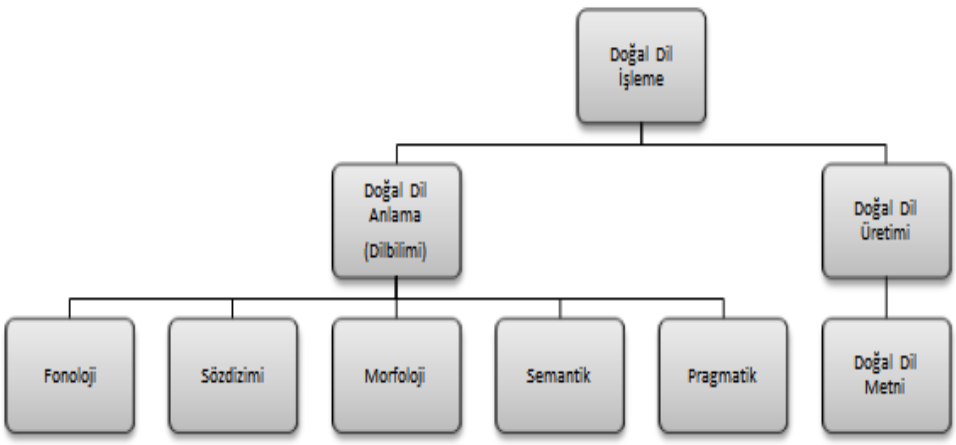
Yapay Zeka'nın alt dalları olarak ayrılan sistemler ise aşağıda listelenmektedir (Wikipedia, 2020).

- Makine Zekâsı
- Yapay Sinir Ağları
- Doğal Dil işleme
- Konuşma Sentezi
- Konuşma Anlama
- Uzman Sistemler
- Örüntü Tanıma
- Genetik Algoritmalar
- Genetik Programlama
- Bulanık Mantık
- Çoklu Örnekle Öğrenme

3.1. Doğal Dil İşleme (Natural Language Processing)

Yapay Zeka'nın bir alt dalı olan ve bu çalışmanın temelini oluşturan Doğal Dil İşleme (DDİ) kavramını açıklamak sonraki süreçleri daha anlamlı hale getirecektir. Doğal Dil İşleme (Natural Language Processing) , bilgisayarların yararlı şeyler yapmak için doğal dili (Türkçe, İngilizce vb.) anlamak ve değiştirmek için nasıl kullanılabileceğini araştıran bir araştırma ve uygulama alanıdır. Doğal Dil İşleme için araştırma yapanlar, insanların dili nasıl anladığı ve kullandıkları hakkında bilgi toplamayı amaçlamaktadır. Bu şekilde bilgisayar sistemlerinin istenen görevleri yerine getirmek için doğal dilleri anlamasını ve teknikler geliştirmesini sağlar (Chowdhury, 2003).

Bir diğer kaynakta Doğal Dil İşleme, insan dilini hesaplamalı olarak temsil etmek ve bu dilleri analiz etmek için kullanılan bir sistem olduğu ifade edilmektedir (Khurana vd., 2017). Bu durum aslında, doğal dillerin bilgisayar veya diğer elektronik cihazlarda temsil edilebilmesini sağlamaya yönelik olan çalışmalar olarak görülebilir. Doğal Dil İşlemenin (DDİ) geniş bir perspektif ile sınıflandırılması aşağıdaki Şekil 3.1'de gösterilmektedir.



Şekil 3.1. DDİ sınıflandırılması

Doğal Dil işleme kavramı insanlığa üzerinde çalışılacak birçok konuyu beraberinde getirmiştir. Bu konular aşağıda sıralanmaktadır (Adalı, 2012).

- Yazım yardımcı araçlarının geliştirilmesi
- Yazım yanlışlarının düzeltilmesi
- Bul ve değiştir
- Basılı bir metni okuma ve okuma yanlışlarını düzeltme
- Bir metnin özetini çıkarma
- Metnin içerdiği bilgiyi çıkarma
- Bilgiye erişim
- Metni anlama
- Bilgisayarla sesli etkileşim
- Bilgisayarın konuşması (metni seslendirme)
- Konuşmayı anlama (konuşmayı metne dönüştürme)
- Soru yanıt dizgeleri
- Yabancı dil okuma yardımcı araçları
- Yabancı dilde yazma yardımcı araçları
- Doğal diller arası çeviri

3.2. Zemberek Kütüphanesi

Çalışmada, Türkçe haber metinlerinin kelimelere ayrıştırma (Tokenization), Türkçe dil bilimi (Morfoloji), ve önemsiz görülen kelimeleri silme (Stop Words) işlemi için Akın kardeşler tarafından geliştirilmiş, Java tabanlı Zemberek Kütüphanesi kullanılmıştır.

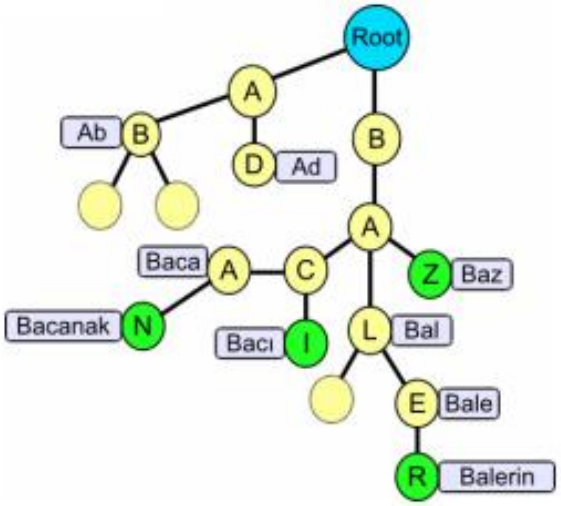
DDİ Zemberek Kütüphanesi ile yazım denetimi, kök bulma (Stemming), morfolojik ayrıştırma, kelime önerme, kelime oluşturma, ASCII karakterleri kullanılarak yazılan sözcükleri dönüştürme ve heceleri çıkarma gibi temel olan DDİ işlemleri yapılabilmektedir (Akın ve Akın, 2007).

3.2.1. DDİ Zemberek kütüphanesinin yapısı

NLP Zemberek Kütüphanesinin yapısı iki ana bölümden oluşmaktadır: Dil yapısı bilgisi ve DDİ işlemleri. Çekirdek kitaplık doğal dil işlemeye özgü algoritmalar içerir ve dil uygulamalarına gerekli araçları sağlar. Çekirdek kütüphane özellikle Türk dilleri için tasarlanmış olsa da, herhangi bir özel dil uygulaması içermez. Bu esnekliği sağlamak için çeşitli yardımcı mekanizmalar ve soyutlamalar kullanılmaktadır. Her dil uygulaması, önceden tanımlanmış dilbilgisi gereksinimlerine uymaktan ve gerekli dil verilerini sağlamaktan sorumludur (Akın ve Akın, 2007). Kütüphanenin açık kaynaklı olması ve Mozilla Kamu Lisansına (MPL) sahip herkesin projeye katkıda bulunabilmesi bir diğer avantajları arasında yer alır (GitHup, 2020).

3.2.2. DDİ Zemberek kütüphanesinin kök ağacı

Zemberek, kök sözlük tabanlı bir ayrıştırıcı kullandığından, bir kök bulma mekanizması gerektirir. Ayrıştırıcı, kök adaylarını bularak ayrıştırma işlemine başlar. DDİ Zemberek kütüphanesinin kullandığı kök ağacı Şekil 3.2'de gösterilmektedir (Akın ve Akın, 2007).



Şekil 3.2. Doğrudan çevrimsel kelime grafiği ağacı

Şekil 3.2'deki ağaçta kökler içeriklerine uygun şekilde yerleştirilir. Örneğin yukarıdaki "Baca" kelimesi sırasıyla B, A, C, A ile etiketlenmiş düğümlerin en

sonuncusuna bağlanmış şekilde yer alır. Dikkat çeken bir diğer nokta da uzun olan kelimeler, fazladan düğüm oluşturmamak adına şekildeki gibi ağaca bağlanır ve böylece bellekten tasarruf edilir. Yukarıda görülen "Bacanak" kelimesi B, A, C, A düğümlerinden sonra gelen N düğümüne bağlanır.

3.2.3. Zemberek ile TRNLTK karşılaştırması

TRNLTK, Ali Ok tarafından ve Zemberek'in kurucuları Akın kardeşlerinde birkaç noktada dâhil olduğu, 2013 yılında geliştirilmiş bir Türk Doğal Dil İşleme kütüphanesidir. Bu kütüphane ile bakımı, genişletmesi ve özelleştirmesi kolay, sağlam bir morfolojik ayrıştırıcı hedeflenmektedir. Açık kaynak kodlu olan ve sadece Türkçe dili için hazırlanan TRNLTK ayrıştırıcı, Zemberek'e bir alternatiftir (Ali Ok, 2020).

TRNLTK'ın avantajları:

- Son derece özelleştirilebilir
- Takılabilir ek grafikleri
- Takılabilir kök bulucular
- Çok tutarlı bir sözlük
- Bilinmeyen kelimeler için kaba kuvvet yöntemi
- Yüksek başarı oranı (kelimelerin% 99'unu ayrıştırabilir)
- Daha az hileli
- Bakımı çok daha kolaydır
- Morfolojik ayrıştırıcı daha fazla ayrıştırma sonuçları sunar

Zemberek'in avantajları:

- Proje daha aktif olarak geliştirildi
- Performans daha iyi
- Daha büyük bir topluluğu var

TRNLTK, birlikte sunduğu ek grafikleri ve kök bulucuları Türk dil biliminin (morfolojisinin) neredeyse tamamını kapsamaya ve çok fazla sonuç üretmesine neden olsa da Zemberek'e göre çok yavaştır (TRNLTK 1250 token /

saniye, Zemberek ise 20000 token / saniye). Bu açıdan Zemberek ayrıştırıcı, bellek ve işlemci sınırlamaları durumunda daha uygundur (GitHub, 2020). Bu sebeple proje de Zemberek Kütüphanesi kullanılmıştır.

3.3. Hata Matrisi (Confusion Matrix)

Hata matrisi, gerçek değerlerin bilinmekte olduğu bir veri seti üzerinde, tahmin değerlerini de dahil ederek sınıflandırma modelinin performansını tanımlamak için sıklıkla kullanılan bir tablodur. Hata matrisi tablosunun bir örneği Çizelge 3.1'de gösterilmektedir.

Çizelge 3.1. Hata matrisi örneği

		Tahmin Edilen	
		Pozitif	**Negatif**
Gerçek	**Pozitif**	Doğru Pozitif	Yanlış Negatif
	Negatif	Yanlış Pozitif	Doğru Negatif

Çizelge 3.1'de yer alan Pozitif ve Negatif iki sınıflı Hata Matrisini daha iyi anlamak için içerdiği tüm kavramların doğru şekilde bilinmesi gerekmektedir. Aşağıda kavramların açıklamasına yer verilmiştir (Dev, 2020).

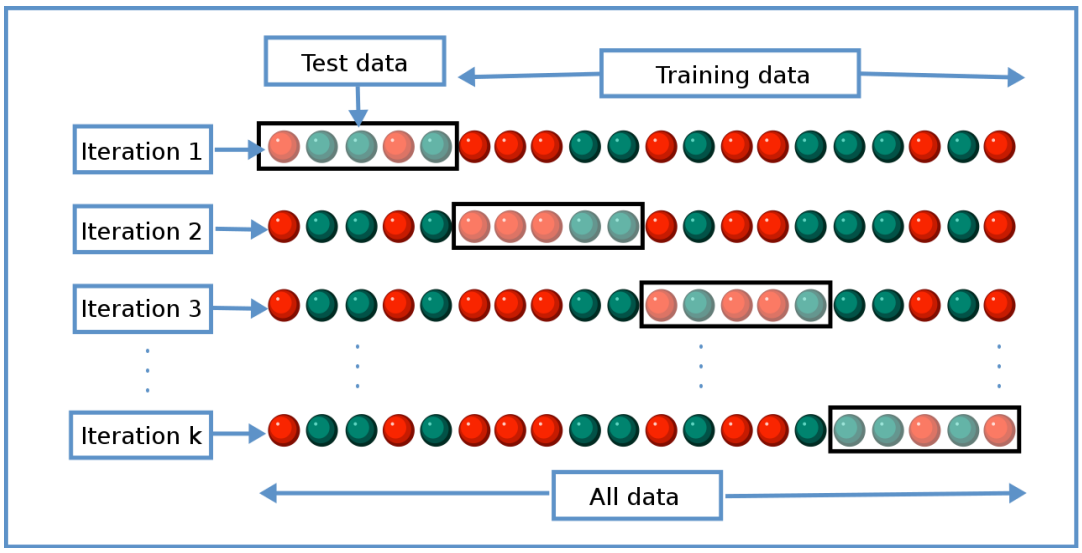
Doğru Pozitif: Gerçek pozitif değerle tahmin değerin eşleşmesi
Doğru Negatif: Gerçek negatif değerle tahmin değerin eşleşmesi
Yanlış Pozitif: Gerçek negatif değerin, tahmin değerinin pozitif olması
Yanlış Negatif: Gerçek pozitif değerin, tahmin değerinin negatif olması
Hata matrisi ile sınıflandırma modelini yorumlamak için Doğruluk (Accuracy), Hatırlama (Recall), Tutarlılık (Precision), F1- Değer, Ağırlıklı Ortalama ve daha fazla sonuç değerleri elde edilebilir.
Doğruluk (Accuracy): Doğru sınıflandırılan örnek sayısının toplam örnek sayısına oranı olarak hesaplanır.
Hatırlama (Recall): Pozitif olan örneklerden, ne kadarının pozitif olarak tahmin edildiğini gösteren bir metriktir.
Tutarlılık (Precision): Tutarlılık çok sınıflı bir karışıklık matrisidir. Pozitif olarak tahmin edilen örneklerden, kaçının gerçekten pozitif olduğunu göstermektedir.

F1-Değer: Precision ve Recall değerlerinin harmonik ortalamasıdır.

Ağırlıklı Ortalama: Her sınıf için elde edilen bir metrik değerinin ortalamasıdır.

3.4. K-Katlamalı Çapraz Doğrulama

K-katlamalı çapraz doğrulama (K- fold cross validation) istatistiksel sonuçları değerlendirmek için kullanılan model doğrulama tekniklerinden biridir (Wikipedia, 2020). K-katlamalı çapraz doğrulama veri kümesinin rastgele "k" tane gruba ayrılması işlemidir. Gruplardan biri test veri seti olarak kullanılırken, geri kalanlar ise eğitim veri seti olarak kullanılır. Her bir grup bu şekilde "k" kadar tekrarlanarak model eğitilir ve diğer grup ile test edilir. Bu şekilde model tüm veriler ile eğitilmiş olacaktır ki bu işlemde modelin doğruluğu için oldukça gereklidir. Örneğin, 10-katlamalı çapraz doğrulama için, veri seti 10 gruba ayrılır ve model 10 kez eğitilip test edilir. Böylece her grup test seti olma şansıda elde edecektir (Medium, 2020). Şekil 3.3'te k-katlamalı çapraz doğrulama işleminin işlem adımları gösterilmektedir.



Şekil 3.3. K-katlamalı çapraz doğrulama (Wikipedia, 2020)

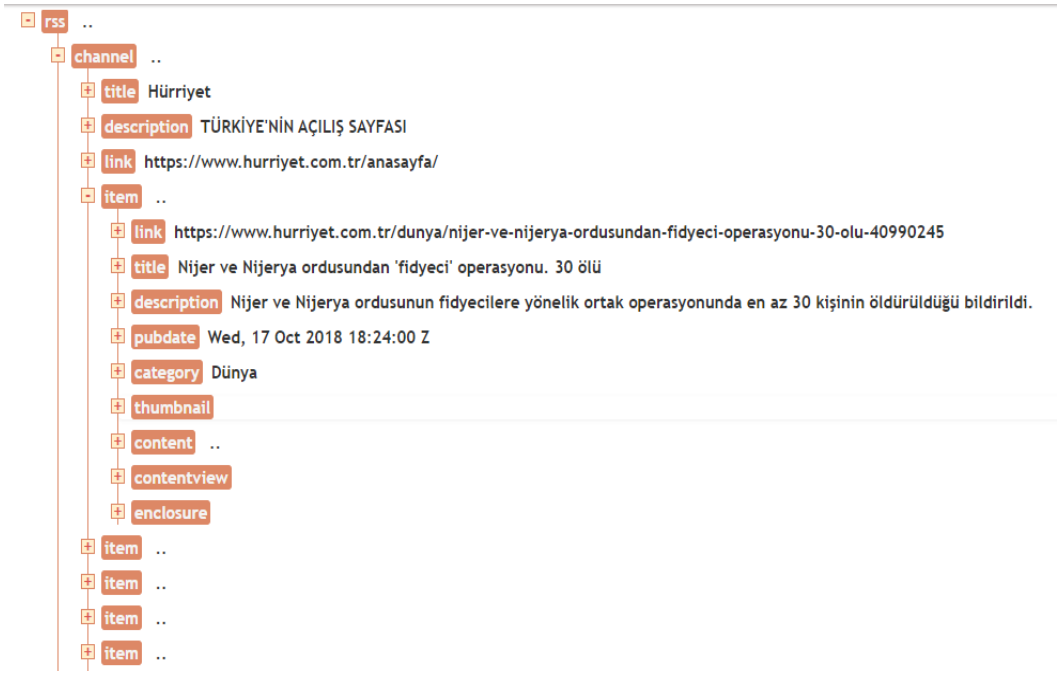
4. YÖNTEM

Çalışma kapsamında bir sözlüğe ihtiyaç duyulmuştur. Çünkü haberlerin yaş grubunun bulunabilmesi için öncelikle ilgili yaş gruplarına ait olan haberlerden belli bir ölçüte göre anlamlı kelimeler seçilip (yaş grubu bilgisi ve kelimenin haberdeki sıklık bilgisi de yer alacak şekilde) bir havuz oluşturulmalıdır. Daha sonra herhangi başka bir haberin sahip olduğu kelimeler, oluşturulan bu havuzdaki kelimeler ile karşılaştırılarak hangi yaş grubundaki kelimelerle daha çok örtüştüğüne karar verilip haberin yaş grubu bilgisinin tahmin edilmesi için bir sözlük kullanılmaktadır.

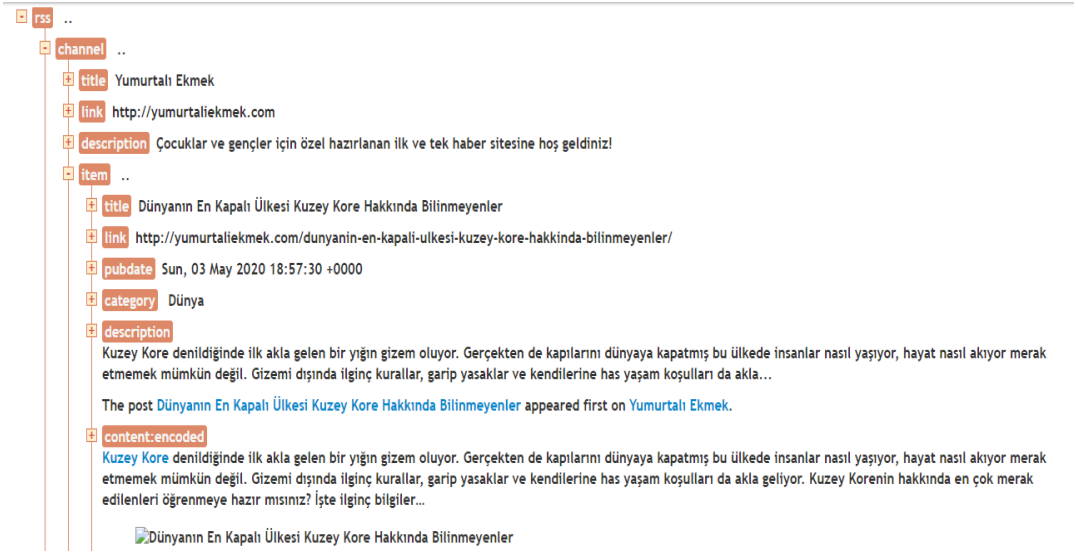
Sözlük 4 aşamada oluşturulmaktadır. Bu aşamalar veri seti oluşturulması, veri ön işleme işlemleri, sözlük oluşturma ve tahminleme olarak sıralanır. Çalışmada Pyhton dili kullanılmıştır. Kullanılan Pyhton Sürümü Python 3.7 ve kullanılan IDE (Integrated Development Environment) Visual Studio 2019'dur.

4.1. Veri Seti

Hazır bir veri seti bulunmadığından, sözlük oluşturmak için kullanmak üzere veriler haber sitesinden temin edilmiştir. Bunun için Türkiye'nin en çok okunan haber sitelerinden Hürriyet (Hürriyet, 2020) ve çocuklara-gençlere özel hazırlanan ilk ve tek haber sitesi olan Yumurtalı Ekmek (Yumurtalı Ekmek, 2020) adlı haber sitelerinin haberleri kullanılmıştır. Yeni eklenen içeriğin kolaylıkla takip edilmesini sağlayan doküman takip sistemi RSS (Really Simple Syndication) kullanılmıştır (Wikipedia, 2020). RSS sistemi üzerinden Hürriyet (Hürriyet RSS, 2020) ve Yumurtalı Ekmek (Yumurtalı Ekmek RSS, 2020) sitelerinin RSS haberleri alınmıştır. İki haber sitesinin de RSS yapısını anlamak ve veri setini doğru şekilde oluşturmak adına Code Beautify adlı sitenin yardımıyla RSS servis yapılarının ağaç görüntüsü üzerinde incelemeler yapılmıştır. Şekil 4.1 ve Şekil 4.2'de görüldüğü üzere haber sitelerinde haberler RSS servislerinde "item" etiketinde (tag) tutulmaktadır. Her "item" bir habere karşılık gelmektedir ve Şekil 4.1 ve Şekil 4.2'de bir haber örneği gösterilmiştir.



Şekil 4.1. Hürriyet haber sitesinin RSS yapısı (Code Beautify, 2020)



Şekil 4.2. Yumurtalı Ekmek haber sitesinin RSS yapısı (Code Beautify, 2020)

RSS servislerinde XML (Extensible Markup Language) dosya biçimi kullanılmıştır. XML dosya biçimi Python'ın "Feedparser" kütüphanesi ile çözümlenmiştir. XML çözümlenmesinden sonra haberin HTML (Hyper Text Markup Language) etiketlerinden temizlenmesi için Python'ın "BeautifulSoup4" kütüphanesi kullanılmıştır. Kullanılan haberlerde çalışma için gerekli olan haberin başlığı, haberin linki, haberin kategorisi, haberin kısa içeriği, haberin

tam içeriği, haberin fotoğraf bilgisi, haberin yayınlandığı tarih ve haberin eklenme tarih bilgisi alınmıştır.

Haberlerin bir veri tabanında tutulması için SQLite ilişkisel bir veri tabanı kullanıldı. Bunun için Python'ın "Sqlite3" kütüphanesi kullanılarak bir veri tabanı oluşturuldu. Oluşturulan veri tabanında Haberler adında bir tablo oluşturuldu. Oluşturulan Haberler tablosuna "Title", "Link", "CategoryName", "Description", "Details", "MediaUrl", "PubDate", "InsertDate" adında 8 adet sütun bilgisi eklendi ve RSS üzerinden alınan haber bilgileri sırasıyla Haberler tablosundaki ilgili alanlara ekleme işlemi yapıldı. Çizelge 4.1'de haberler tablosunun ayrıntılı bilgisi bulunmaktadır.

Çizelge 4.1. Haberler tablosunun içeriği

Haberler Tablosu		
Sütun Adı	**Veri Tipi**	**Açıklama**
Title	Text	Habere ait başlık bilgisi
Link	Text	Haberin detayına ait link bilgisi
CategoryName	Text	Haberin ait olduğu kategori bilgisi
Description	Text	Habere ait özet bilgisi
Details	Text	Habere ait detay bilgisi
MediaUrl	Text	Habere ait resim bilgisi
PubDate	Date	Haberin yayınlandığı tarih bilgisi
InsertDate	Date	Haberin tabloya eklenme tarih bilgisi

Türkiye'nin en çok okunan haber sitelerinden Hürriyet ve çocuklara-gençlere özel hazırlanan ilk ve tek haber sitesi Yumurtalı Ekmek adlı haber sitesi kullanılarak, toplam da 3925 haberin SQLite ile veri tabanına kaydedilmesinden sonra veri tabanındaki kayıtlar bir tablolama programına (Excel) aktarıldı. Çizelge 4.2'de haberlerin haber sitelerine göre sayısal dağılımı gösterilmektedir.

Çizelge 4.2. Haberlerin haber sitelerine göre sayısal dağılımı

Haber Sitesi	**Toplam Haber Sayısı**
Yumurtalı Ekmek	1313
Hürriyet	2612
Toplam	3925

Excel de yer alan haberlerin sütun bilgilerine ek olarak Yaş sütunu eklendi. Yaş sütunu çalışmada gönüllü olarak yer alan Rehberlik ve Psikolojik Danışman tarafından Havighurst'ün gelişim kuramının uyarlanmış yaş gruplarına göre (Çocukluk, Ergenlik, Yetişkinlik) etiketlendirilmesi yapıldı. Etiketleme işlemi de tamamlandıktan sonra veri seti hazır hale getirildi. Oluşturulan veri setindeki haberlerin sınıflandırılmış halinin yaş gruplarına göre sayısal dağılımı Çizelge 4.3'te görülmektedir.

Çizelge 4.3. Haberlerin yaş gruplarına göre sayısal dağılımı

Yaş Grubu	Haber Sayısı
Çocukluk (6-13 yaş)	1313
Ergenlik (13-18)	1189
Yetişkinlik (18+)	1423

Oluşturulan veri setinin bir görüntüsü Şekil 4.3'te gösterilmektedir. Yaş sütununda yer alan "1" Çocukluk yaş grubunu, "2" Ergenlik yaş grubunu, "3" ise Yetişkinlik yaş grubunu temsil etmektedir.

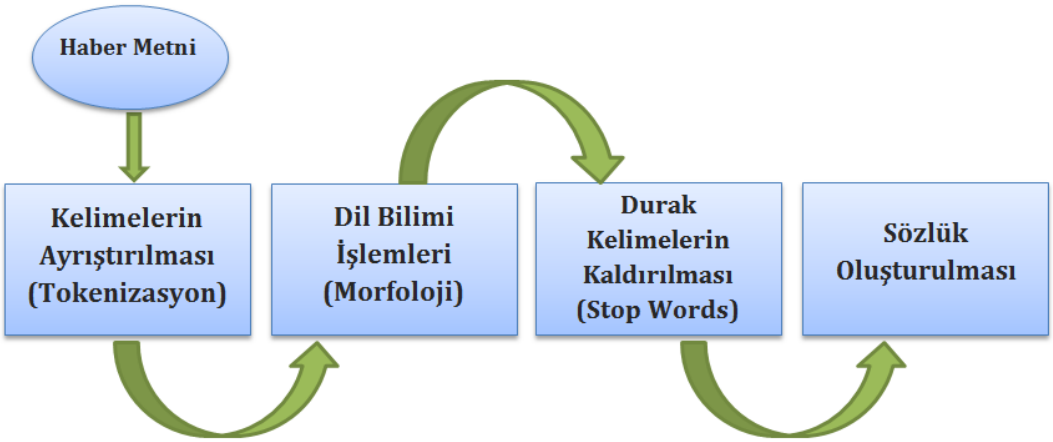
Title	Description	Details	Link	MediaUrl	CategoryName	PubDate	InsertDate	Yaş
Bitter Çikolatanın 10 Muhteşem Faydası	Çikolatalar bilindiklerinin aksine oldukça faydalı bir besindir. Özellikle bitter çikolatanın içerdiği bileşenler nedeniyle birçok hastalığa da iyi geldiği biliniyor.	Çikolatalar bilindiklerinin aksine oldukça faydalı bir besindir. Özellikle bitter çikolatanın içerdiği bileşenler nedeniyle birçok hastalığa da iyi geldiği biliniyor. Bitter çikolatanın	http://www.hurriyet.com.tr/mahmure/galeri-bitter-cikolatani	https://i4.hurimg.com/i/hurriyet/75/620x350/5d7941d545d2a023	Mahmure	2018-03-20 10:40:00.000	01.11.2019 11:54	1
10 yılda 280 kostüm ve 47 çift ayakkabı sattı	Ebru Gündeş, 2009 yılında hayata geçirdiği "Ebru Gündeş Satıyor" projesiyle birçok vakıf ve derneğe önemli kaynak sağladı.	Ebru Gündeş, 2009 yılında hayata geçirdiği "Ebru Gündeş Satıyor" projesiyle birçok vakıf ve derneğe önemli kaynak sağladı.Ebru Gündeş, 2009 yılında hayata	http://www.hurriyet.com.tr/galeri-10-yilda-280-	https://i4.hurimg.com/i/hurriyet/75/620x350/5da2d3c9	Kelebek	2019-10-13 10:36:37.000	01.11.2019 11:54	2
PKK DEAŞ'tan daha kötü bir tehdit	Beyaz Saray'da İtalya Cumhurbaşkanı Sergio Mattarella'yla görüşmesi sonrası kameralar karşısına geçen ABD Başkanı Donald Trump, terör örgütü PKK'nın DEAŞ'tan daha kötü olduğunu söyledi.	Beyaz Saray'da İtalya Cumhurbaşkanı Sergio Mattarella'yla görüşmesi sonrası kameralar karşısına geçen ABD Başkanı Donald Trump, terör örgütü PKK'nın DEAŞ'tan daha kötü	http://www.hurriyet.com.tr/dunya/pkk-deastan-daha-	https://i4.hurimg.com/i/hurriyet/75/620x350/5da79a130f254420	Dünya	2019-10-16 22:00:00.000	01.11.2019 11:54	3

Şekil 4.3. Çalışmada kullanılan veri seti

4.2. Veri Ön İşleme

Veri ön işleme, herhangi bir veri kümesindeki ilk işlemdir. Gerçek veri işleme başlamadan önce yapılan tüm işlemlerden (verinin doğruluğu, eksiklerinin giderilmesi, normalleştirme gibi işlemler) oluşur. Bu işlem ile veriler üzerinde herhangi bir analiz yapılmasını engelleyebilecek veri problemlerini çözmek, verinin doğasını anlamak, daha anlamlı bir veri analizi yapmak ve belirli bir veri kümesinden daha anlamlı bilgiler elde etmek gibi faydalar sağlanabilir (İlhan, 2001).

Bu doğrultuda veri ön işlemenin bir kısmı olan HTML ve XML etiketlerinin temizlenmesi veri seti başlığı altında anlatıldığı üzere sonraki adımlar olan kelimelere ayrıştırma, dil bilimi işlemleri ve durak kelimelerin kaldırılması adımlarının öncelikle tanımları, daha sonra çalışmada ne amaçla kullanıldıklarına bu başlıkta yer verilecektir. Ayrıca veri ön işleme çalışma adımları Şekil 4.4'te gösterilmektedir.



Şekil 4.4. Veri ön işleme adımları

3925 haber içinden her yaş grubuna eşit olarak 1000 haber alındı. Toplamda 3000 haber alınarak veri setinin %70'lik kısmı eğitim için %30'luk kısmı ise sınama amaçlı kullanılmak üzere ayrılmıştır.

4.2.1. Kelimelere ayrıştırma (Tokenizasyon)

Kelimelere ayrıştırma (Tokenizasyon), belgelerin yalın halinden, hesaplamalı dilbilimi uygulamaları için uygun olan hesaplama birimlerine (kelime), hesaplama simgelerine dönüştürülmesi ve cümle sonlarının tespitidir (Dinçer, 2004).

Kelimelere ayrıştırma işlemi gerçekleştirilirken NLP Zemberek Kütüphanesinin "Tokenizer" sınıfı ile haber kelimelere ayrıştırılmıştır. Bunun yanı sıra metin içinden "Boşluk, yeni satır, noktalama işareti, roma rakamı, numara, yüzdelik ifade, zaman, tarih, URL, E-mail, HashTag, Mention, MetaTag, Emoji, Emoticon, Bilinmeyen" olan kelimeler temizlenmiştir. Ayrıştırma işlemi yapılmadan önceki örnek haber metinleri Çizelge 4.4'te kelimelere ayrıştırma işleminden sonra elde edilen sonuçlar ise Çizelge 4.5'te gösterilmektedir.

Çizelge 4.4. İşlenmemiş haber metinleri

Yaş Grubu	**İşlenmemiş Haber Metinleri**
Çocukluk	Milli Eğitim Bakanlığı, SBS sonucuna göre öğrenci alan liselerin kontenjanlarını yüzde 30 artırdı. Milli Eğitim Bakanlığı, Seviye Belirleme Sınavı (SBS) sonucuna göre öğrenci alan fen ve sosyal bilimler liseleri ile her türdeki Anadolu liselerinin kontenjanlarını yüzde 30 artırdı Bugün açıklanan sonuçlara göre, tercihler 15 Temmuz pazartesi günü başlayacak. Yeni düzenlemelerle daha fazla öğrencinin Fen ve Sosyal Bilimler liseleri ile her türdeki Anadolu liselerinde öğrenim görmesi amaçlanıyor. Geçen yıl fen liselerinin 9. sınıf kontenjanı 12 bin 376 iken, kapasitede sağlanan artışla 2013-2014 eğitim-öğretim yılında 15 bin 780 öğrencinin bu okulların 9. sınıflarına yerleşebilmelerine imkan sağlandı. Ayrıca Anadolu liselerinin 9. sınıf kontenjanı 226 bin 900'den 314 bin 110'a, Anadolu öğretmen liselerinin kontenjanı 29 bin 780'den 31 bin 200'e, sosyal bilimler liselerinin kontenjanı 2 bin 522'den 2 bin 782'ye, Anadolu imam hatip liselerinin kontenjanı 42 bin 144'ten 64 bin 170'e, Anadolu türü mesleki teknik eğitim okullarının kontenjanı ise 139 bin 513'ten 184 bin 707'ye çıkarıldı.
Ergenlik	Fransa Ligue 1'in 10'uncu hafta maçında milli futbolcular Yusuf Yazıcı ve Mehmet Zeki Çelik'in formasını giydiği Lille deplasmanda konuk olduğu Toulouse'a 2-1 mağlup oldu. Ev sahibi ekibe galibiyeti getiren golleri 58'inci dakikada Yaya Sanogo ve 66'ncı dakikada Max Gradel kaydederken, konuk

	takımın tek golü 90'ıncı dakikada Fonte'den geldi. Milli futbolcu Yusuf Yazıcı müsabakaya 77'nci dakikada dahil olurken takımının attığı golde asisti yapan isim oldu. Mehmet Zeki Çelik ise sakatlığı nedeniyle mücadelede forma giyemedi.
Yetişkinlik	1 günde 10 kilo eroin satmışlar. Ankara 3'üncü Ağır Ceza Mahkemesi'nde devam eden dava dosyasında, itirafçı olan sanık Ö.D'nin ifadesine göre, çete Başkent sokaklarında torbacılara kilosu 30 bin liradan eroin sattı. Torbacılar da tahmini olarak günde 10 kilo eroini sokaklarda pazarladı. OPERASYONLA YAKALANMIŞLARDIBaşkent'te geçtiğimiz yıl uyuşturucu satıcılarına yönelik, helikopter destekli düzenlenen 'kalkan' operasyonunda aralarında örgüt lideri Muzaffer Arkaç'ın da yer aldığı 3'ü polis 111 kişi yakalanmış, 67 kilogram eroin, 38 bin 580 uyuşturucu hap, 21 ruhsatsız tabanca, 14 tüfek ile aralarında lüks cip ve otomobillerin de yer aldığı 21 araç ele geçirilirken, uyuşturucu satışından elde edildiği belirlenen 120 bin 918 liraya el konulmuştu.

Çizelge 4.5. Kelimelere ayrıştırma (Tokenizasyon) işlemi

Yaş Grubu	**Kelimelere Ayrıştırma (Tokenizasyon)**
Çocukluk	['Milli', 'Eğitim', 'Bakanlığı', 'SBS', 'sonucuna', 'göre', 'öğrenci', 'alan', 'yüzde', 'artırdı', 'Milli', 'Eğitim', 'Bakanlığı', 'Seviye', 'Belirleme', 'Sınavı', 'SBS', 'sonucuna', 'göre', 'öğrenci', 'alan', 'fen', 've', 'sosyal', 'bilimler', 'liseleri', 'ile', 'her', 'türdeki', 'Anadolu', 'liselerinin', 'kontenjanlarını', 'yüzde', 'artırdı', 'Bugün', 'açıklanan', 'sonuçlara', 'göre', 'tercihler', 'Temmuz', 'pazartesi', 'günü', 'başlayacak', 'Yeni', 'düzenlemelerle', 'daha', 'fazla', 've', 'Sosyal', 'Bilimler', 'liseleri', 'ile', 'her', 'türdeki', 'Anadolu', 'liselerinde', 'öğrenim', 'görmesi', 'amaçlanıyor', 'Geçen', 'yıl', 'fen', 'liselerinin', 'sınıf', 'kontenjanı', 'bin', 'iken', 'kapasitede', 'sağlanan', 'artışla', 'eğitim', 'öğretim', 'yılında', 'bin', 'öğrencinin', 'bu', 'okulların', 'sınıflarına', 'yerleşebilmelerine', 'imkan', 'sağlandı', 'Ayrıca', 'Anadolu', 'liselerinin', 'sınıf', 'kontenjanı', 'bin', 'bin', 'Anadolu', 'öğretmen', 'liselerinin', 'kontenjanı', 'bin', 'bin', 'sosyal', 'bilimler', 'liselerinin', 'kontenjanı', 'bin', 'bin', 'Anadolu', 'imam', 'hatip', 'liselerinin', 'kontenjanı', 'bin', 'bin', 'Anadolu', 'türü', 'mesleki', 'teknik', 'eğitim', 'okullarının', 'kontenjanı', 'ise', 'bin', 'bin', 'çıkarıldı']
Ergenlik	['Fransa', 'Ligue', 'uncu', 'hafta', 'maçında', 'milli', 'futbolcular', 'Yusuf', 'Yazıcı', 've', 'Mehmet', 'Zeki', 'Çelik'in', 'formasını', 'giydiği', 'Lille', 'deplasmanda', 'konuk', 'olduğu', 'Toulouse'a', 'mağlup', 'oldu', 'Ev', 'sahibi', 'ekibe', 'galibiyeti', 'getiren', 'golleri', 'dakikada', 'Yaya', 'Sanogo', 've', 'dakikada', 'Max', 'Gradel', 'kaydederken', 'konuk', 'takımın', 'tek', 'golü', 'dakikada', 'Fonte'den', 'geldi', 'Milli', 'futbolcu', 'Yusuf', 'Yazıcı', 'müsabakaya', 'dakikada', 'dahil', 'olurken', 'takımının', 'attığı', 'golde', 'asisti', 'yapan', 'isim', 'oldu', 'Mehmet', 'Zeki', 'Çelik', 'ise',

	'sakatlığı', 'nedeniyle', 'mücadelede', 'forma', 'giyemedi']
Yetişkinlik	['günde', 'kilo', 'eroin', 'satmışlar', 'Ankara', 'Ağır', 'Ceza', 'Mahkemesi'nde', 'devam', 'eden', 'dava', 'dosyasında', 'itirafçı', 'olan', 'sanık', 'Ö.D'nin', 'ifadesine', 'göre', 'sokaklarında', 'torbacılara', 'kilosu', 'bin', 'liradan', 'eroin', 'sattı', 'Torbacılar', 'da', 'tahmini', 'olarak', 'günde', 'kilo', 'eroini', 'sokaklarda', 'pazarladı', 'OPERASYONLA', 'YAKALANMIŞLARDI', 'Başkent'te', 'geçtiğimiz', 'yıl', 'uyuşturucu', 'satıcılarına', 'yönelik', 'helikopter', 'destekli', 'düzenlenen', 'kalkan', 'aralarında', 'örgüt', 'lideri', 'Muzaffer', 'Arkaç'ın', 'da', 'yer', 'aldığı', 'polis', 'kişi', 'yakalanmış', 'kilogram', 'eroin', 'bin', 'uyuşturucu', 'hap', 'ruhsatsız', 'tabanca', 'tüfek', 'ile', 'aralarında', 'lüks', 'cip', 've', 'otomobillerin', 'de', 'yer', 'aldığı', 'araç', 'ele', 'geçirilirken', 'uyuşturucu', 'satışından', 'elde', 'edildiği', 'belirlenen', 'bin', 'liraya', 'el', 'konulmuştu']

4.2.2. Dil bilimi işlemleri (Morfoloji)

Dil bilimi işlemleri (Morfoloji), kelimelere ayrıştırma (Tokenizayon) ile ayrılmış her bir kelimenin kök halinin elde edilmesidir. Dil bilimi işlemleri Stemming ve Lemmatization olmak üzere iki kısma ayrılır.

- Stemming: Bir kelimede yer alan ön ekleri ve son ekleri keserek kelime kökünü bulmaya çalışır (Medium, 2020).

- Lemmatization: Kelimelerin morfolojik analizlerini esas alır. Böylelikle, algoritmanın kelimenin kökünü bulması için detaylandırılmış bir sözlüğe ihtiyacı vardır (Medium, 2020).

Örneğin "kitabımız" kelimesinin Stemming ile kökü "kitab" olarak bulunurken Lemmatization ile "kitap" olarak bulunur.

Dil bilimi işlemi sırasında öncelikle kelimelerin hepsi küçük harfe çevrilerek metin birimleri normalleştirilmiştir. Metin birimleri normalleştirilen kelimeler NLP Zemberek Kütüphanesinin "Morphology" sınıfı ile Lemmatization kök bulma tipi kullanılarak kelimelere ayrıştırma (Tokenizasyon) işleminde elde edilen kelimelerin kökü bulunur. Sözlük oluşturma işleminde sözlüğe hangi kelimelerin eklenmesi kararına kolaylık sağlaması açısından ulaşılan her kök formunun tipide (sıfat, zarf, fiil vb.) etiketlenmiştir. Bu işleme POS (Part-of-

Speech) denmektedir. POS'u bilinmeyen ve belirsiz olan kelimeler bu çalışmaya dahil edilmemiştir. Çizelge 4.5'deki durumun devamı olarak dil bilimi işlemlerinden (Morfoloji) sonra elde edilmiş sonuç Çizelge 4.6'da gösterilmektedir.

Çizelge 4.6. Dil bilimi işlemleri (Morfoloji)

Yaş Grubu	Dil Bilimi İşlemleri (Morfoloji)
Çocukluk	['eğitim', 'bakanlık', 'sbs', 'sonuç', 'göre', 'öğrenci', 'yüzde', 'milli', 'eğitim', 'bakanlık', 'sevi', 'sınav', 'sbs', 'sonuç', 'göre', 'öğrenci', 'fen', 've', 'sosyal', 'bilim', 'lise', 'ile', 'her', 'anadolu', 'lise', 'kontenjan', 'yüzde', 'bugün', 'sonuç', 'göre', 'tercih', 'temmuz', 'pazartesi', 'gün', 'yeni', 'daha', 'fazla', 've', 'sosyal', 'bilim', 'lise', 'ile', 'her', 'anadol', 'lise', 'öğrenim', 'geçen', 'yıl', 'fen', 'lise', 'sınıf', 'kontenjan', 'bin', 'iken', 'kapasite', 'art', 'eğitim', 'öğretim', 'yıl', 'bin', 'öğrenci', 'bu', 'okul', 'sınıf', 'imkan', 'ayrıca', 'anadolu', 'lise', 'sınıf', 'kontenjan', 'bin', 'bin', 'anadol', 'öğretmen', 'lise', 'kontenjan', 'bin', 'bin', 'sosyal', 'bilim', 'lise', 'kontenjan', 'bin', 'bin', 'anadol', 'imam', 'hatip', 'lise', 'kontenjan', 'bin', 'bin', 'anadol', 'tür', 'mesleki', 'teknik', 'eğitim', 'okul', 'kontenjan', 'bin', 'bin']
Ergenlik	['fransa', 'ligue', 'un', 'hafta', 'maç', 'milli', 'futbol', 'yusuf', 'yazı', 've', 'mehmet', 'zeki', 'çelik', 'forma', 'lille', 'deplasman', 'konuk', 'mağlup', 'ev', 'sahip', 'ekip', 'galibiyet', 'gol', 'dakika', 'yaya', 'sanogo', 've', 'dakika', 'max', 'konuk', 'takım', 'tek', 'gol', 'dakika', 'fonte', 'milli', 'futbol', 'yusuf', 'yazı', 'müsabaka', 'dakika', 'dahil', 'takım', 'at', 'gol', 'asist', 'isim', 'mehmet', 'zeki', 'çelik', 'ise', 'sakat', 'neden', 'mücadele', 'forma']
Yetişkinlik	['gün', 'kilo', 'eroin', 'ankara', 'ağır', 'ceza', 'mahkeme', 'devam', 'dava', 'dosya', 'itiraf', 'sanık', 'ifade', 'göre', 'sokak', 'torba', 'kilo', 'bin', 'lira', 'eroin', 'torba', 'da', 'tahmini', 'gün', 'kilo', 'eroin', 'sokak', 'operasyon', 'yıl', 'uyuşturucu', 'yönelik', 'helikopter', 'kalkan', 'ara', 'örgüt', 'lider', 'muzaffer', 'arkaç', 'da', 'yer', 'al', 'polis', 'kişi', 'kilogram', 'eroin', 'bin', 'uyuşturucu', 'hap', 'tabanca', 'tüfek', 'ile', 'ara', 'cip', 've', 'otomobil', 'de', 'yer', 'al', 'araç', 'el', 'uyuşturucu', 'el', 'bin', 'lira', 'el']

4.2.3. Durak kelimelerinin kaldırılması (Stop Words)

Durak kelimelerinin kaldırılması, anlamsız kelimeleri silme (Remove Stop Words) işlemi, tek başına anlam ifade etmeyen her kelimenin kelime listesinden çıkarılmasıdır.

Veri ön işlemenin son basamağı olan durak kelimelerin kaldırılması işleminde Zemberek kütüphanesinin "RemoveStopWords" listesi ve bunun dışında 2 farklı kaynakta yer alan durak kelimeler projede tek bir listede birleştirilerek kullanılmıştır (GitHup, 2020). Bu üç kaynakta yer alan durak kelimelerine ek olarak projeye ayrıca aylar (Ocak, Şubat vb.), günler (Pazartesi, Salı vb.) ve iller (Adana, Adıyaman vb.) olarak manuel eklemeler yapılarak durak kelimeler listesi genişletilmiştir. Bu genişletilmiş listede yer alan her kelime dil bilimi işlemleri (Morfoloji) aşamasının sonunda elde edilen kelimeler içinde bulunuyor ise kaldırılmıştır. Çizelge 4.6'daki durumun devamı olarak durak kelimelerinin kaldırılma işleminden sonra elde edilen sonuç Çizelge 4.7'de gösterilmektedir.

Çizelge 4.7. Durak kelimelerin kaldırılma işlemi

Yaş Grubu	**Durak Kelimelerin Kaldırılması**
Çocukluk	['eğitim', 'bakanlık', 'sbs', 'sonuç', 'öğrenci', 'yüzde', 'milli', 'eğitim', 'bakanlık', 'sevi', 'sınav', 'sbs', 'sonuç', 'öğrenci', 'fen', 'sosyal', 'bilim', 'lise', 'anadolu', 'lise', 'kontenjan', 'yüzde', 'bugün', 'sonuç', 'tercih', 'gün', 'yeni', 'fazla', 'sosyal', 'bilim', 'lise', 'anadol', 'lise', 'öğrenim', 'geçen', 'yıl', 'fen', 'lise', 'sınıf', 'kontenjan', 'bin', 'kapasite', 'art', 'eğitim', 'öğretim', 'yıl', 'bin', 'öğrenci', 'okul', 'sınıf', 'imkan', 'anadolu', 'lise', 'sınıf', 'kontenjan', 'bin', 'bin', 'anadol', 'öğretmen', 'lise', 'kontenjan', 'bin', 'bin', 'sosyal', 'bilim', 'lise', 'kontenjan', 'bin', 'bin', 'anadol', 'imam', 'hatip', 'lise', 'kontenjan', 'bin', 'bin', 'anadol', 'tür', 'mesleki', 'teknik', 'eğitim', 'okul', 'kontenjan', 'bin', 'bin']
Ergenlik	['fransa', 'ligue', 'un', 'hafta', 'maç', 'milli', 'futbol', 'yusuf', 'yazı', 'mehmet', 'zeki', 'çelik', 'forma', 'lille', 'deplasman', 'konuk', 'mağlup', 'ev', 'sahip', 'ekip', 'galibiyet', 'gol', 'dakika', 'yaya', 'sanogo', 'dakika', 'max', 'konuk', 'takım', 'gol', 'dakika', 'fonte', 'milli', 'futbol', 'yusuf', 'yazı', 'müsabaka', 'dakika', 'takım', 'at', 'gol', 'asist', 'isim', 'mehmet', 'zeki', 'çelik', 'sakat', 'mücadele', 'forma']
Yetişkinlik	['gün', 'kilo', 'eroin', 'ağır', 'ceza', 'mahkeme', 'devam', 'dava', 'dosya', 'itiraf', 'sanık', 'ifade', 'sokak', 'torba', 'kilo', 'bin', 'lira', 'eroin', 'torba', 'tahmini', 'gün', 'kilo', 'eroin', 'sokak', 'operasyon', 'yıl', 'uyuşturucu', 'yönelik', 'helikopter', 'kalkan', 'ara', 'örgüt', 'lider', 'muzaffer', 'arkaç', 'yer', 'al', 'polis', 'kişi', 'kilogram', 'eroin', 'bin', 'uyuşturucu', 'hap', 'tabanca', 'tüfek', 'ara', 'cip', 'otomobil', 'yer', 'al', 'araç', 'el', 'uyuşturucu', 'el', 'bin', 'lira', 'el']

4.3. Sözlük Oluşturma

En son elde edilen kelime listesinde var olan her kelime sözlük için anlamlı durumda değildir. Bu kelime listesinden anlamlı olan kelimeleri sözlüğe eklemek adına birkaç işlem yapılır.

4.3.1. Terim Frekansı

Dokümanlarda en önemli özellik olan terimlerin (çoğunlukla kelime yerine kullanılır) önemi ve taşıdığı bilgi belirlenirken çeşitli metrikler kullanılır. Bunlardan en önemlisi Terim Frekansı (Term Frequency) metriğidir.

Terim Frekansı (TF), metin madenciliğinde en yaygın kullanılan sayısal gösterimdir ve terim sıklığı anlamına gelmektedir. Kısaca, TF değeri bir terimin bir dokümanda görünme sayısını ifade etmektedir (Binici, 2018).

Çizelge 4.7'de veri ön işlemeden geçtikten sonra en son kalan kelimelerin aşağıdaki Çizelge 4.8'de TF değerleri görülmektedir. Bu işlem her haber için yapılmaktadır.

Çizelge 4.8. Terim Frekanslarının bulunması

Yaş Grubu	**Terim Frekansı**
Çocukluk	{'eğitim': 4, 'bakanlık': 2, 'sbs': 2, 'sonuç': 3, 'öğrenci': 3, 'yüzde': 2, 'milli': 1, 'sevi': 1, 'sınav': 1, 'fen': 2, 'sosyal': 3, 'bilim': 3, 'lise': 9, 'anadolu': 2, 'kontenjan': 7, 'bugün': 1, 'tercih': 1, 'gün': 1, 'yeni': 1, 'fazla': 1, 'anadol': 4, 'öğrenim': 1, 'geçen': 1, 'yıl': 2, 'sınıf': 3, 'bin': 12, 'kapasite': 1, 'art': 1, 'öğretim': 1, 'okul': 2, 'imkan': 1, 'öğretmen': 1, 'imam': 1, 'hatip': 1, 'tür': 1, 'mesleki': 1, 'teknik': 1}
Ergenlik	{'fransa': 1, 'ligue': 1, 'un': 1, 'hafta': 1, 'maç': 1, 'milli': 2, 'futbol': 2, 'yusuf': 2, 'yazı': 2, 'mehmet': 2, 'zeki': 2, 'çelik': 2, 'forma': 2, 'lille': 1, 'deplasman': 1, 'konuk': 2, 'mağlup': 1, 'ev': 1, 'sahip': 1, 'ekip': 1, 'galibiyet': 1, 'gol': 3, 'dakika': 4, 'yaya': 1, 'sanogo': 1, 'max': 1, 'takım': 2, 'fonte': 1, 'müsabaka': 1, 'at': 1, 'asist': 1, 'isim': 1, 'sakat': 1, 'mücadele': 1}
Yetişkinlik	{'gün': 2, 'kilo': 3, 'eroin': 4, 'ağır': 1, 'ceza': 1, 'mahkeme': 1, 'devam': 1, 'dava': 1, 'dosya': 1, 'itiraf': 1, 'sanık': 1, 'ifade': 1, 'sokak': 2, 'torba': 2, 'bin': 3, 'lira': 2, 'tahmini': 1, 'operasyon': 1,

	'yıl': 1, 'uyuşturucu': 3, 'yönelik': 1, 'helikopter': 1, 'kalkan': 1, 'ara': 2, 'örgüt': 1, 'lider': 1, 'muzaffer': 1, 'arkaç': 1, 'yer': 2, 'al': 2, 'polis': 1, 'kişi': 1, 'kilogram': 1, 'hap': 1, 'tabanca': 1, 'tüfek': 1, 'cip': 1, 'otomobil': 1, 'araç': 1, 'el': 3}

4.3.2. Eşik değeri bulma

Her habere ait kelimelerin terim frekansları bulunduktan sonra haber bazında anlamlı kelimeleri seçmek üzere bir eşik değeri uygulanmıştır. Bu eşik değeri Formül 4.1'deki gibi ilgili haberdeki her kelimenin TF değerleri toplanarak haberde geçen toplam kelime sayısına bölünmesiyle elde edilmektedir. Böylelikle hesaplanan ortalama değerine eşit veya ortalama değerinden büyük olan TF değerli kelimeler seçilerek, sözlükteki uygun yaş grubunda ilk kez geçiyorsa sıklık değeri "1" olarak (bir haberde görüldü anlamında) eklenir, daha önce geçmiş ise sıklık değeri "1" artırılarak mevcut kayıt güncellenir. Bu işlem tüm eğitim setindeki haberler için tek tek uygulanır.

$$\text{Ortalama} = \frac{\sum \text{Habere ait her kelimenin TF değeri}}{\text{Haberdeki toplam kelime sayısı}} \qquad (4.1)$$

Çizelge 4.8'de örnekleri verilen haberler için eşik değeri formülü uygulandığında ortalamalar sırasıyla 2.29, 1.44, 1.45'tir. Bu sebeple haber bazında TF değeri ortalamadan büyük ve eşit olan kelimeler sözlüğe eklenir. Çizelge 4.9'da bu durum gösterilmektedir.

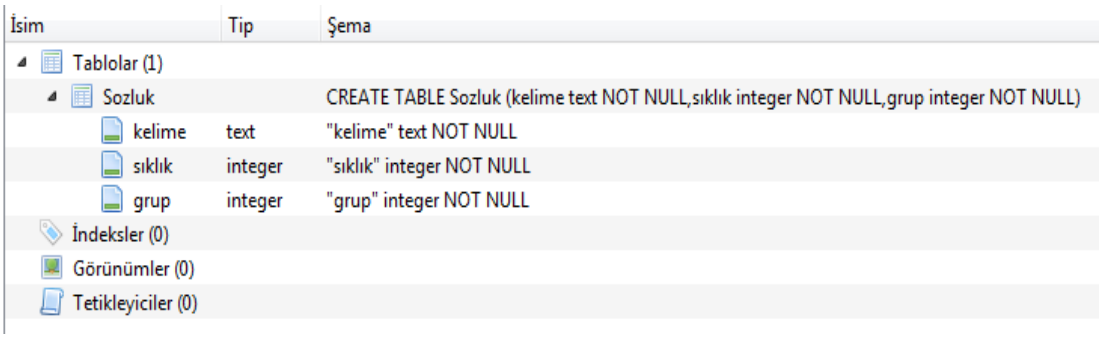
Çizelge 4.9. Terimlerin sözlüğe eklenmesi

Yaş Grubu	**Ortalamadan Büyük Olup Sözlüğe Eklenen Kelimeler**	**Sıklık**
Çocukluk	'eğitim': 4	1
	'sonuç': 3	1
	'öğrenci': 3	1
	'sosyal': 3,	1
	'bilim': 3	1
	'lise': 9	1
	'kontenjan': 7	1
	'anadol': 4	1
	'sınıf': 3	1
	'bin': 12	1

	'okul': 2	1
Ergenlik	'milli': 2	1
	'futbol': 2	1
	'yusuf': 2	1
	'yazı': 2	1
	'mehmet': 2	1
	'zeki': 2	1
	'çelik': 2	1
	'forma': 2	1
	'konuk': 2	1
	'gol': 3	1
	'dakika': 4	1
	'takım': 2	1
Yetişkinlik	'gün': 2	1
	'kilo': 3	1
	'eroin': 4	1
	'sokak': 2	1
	'torba': 2	1
	'bin': 3	1
	'lira': 2	1
	'uyuşturucu': 3	1
	'ara': 2,	1
	'yer': 2	1
	'al': 2	1
	'el': 3	1

4.3.3. Sözlük

Sözlük oluşturulurken aynı yaş grubu için başka bir haberde TF'i eşik değerine eşit veya büyük olup sözlüğe eklenecek kelime aynı ise (daha önce sözlüğe eklenmiş ise) sözlükte tutulan "Sıklık" değeri "1" artırılır. Böylece kelimelerin yaş gruplarına uygun şekilde dağılımı görülmektedir. Tamamlanan sözlüğün ufak bir kısmı Çizelge 4.10'da bulunmaktadır. Elde edilen kelime, kelimenin yaş bilgisi ve sıklık bilgisi ile "Sözlük" adında oluşturulan tablo veri tabanında tutulmaktadır (Şekil 4.5).

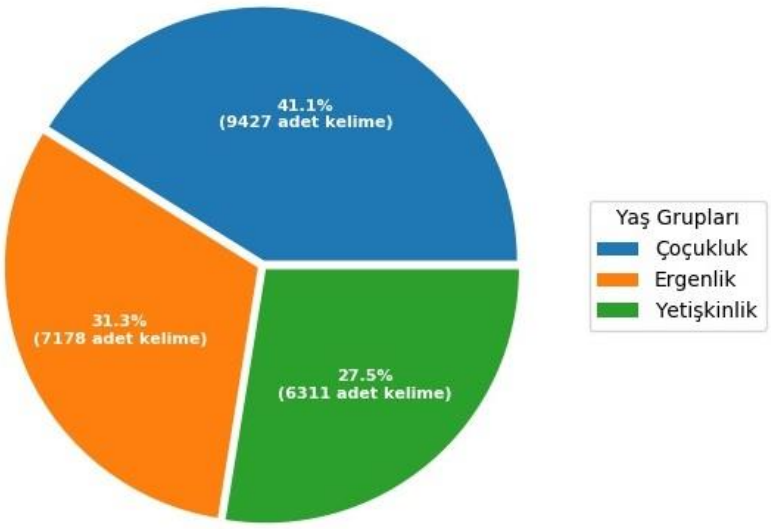


Şekil 4.5. Sözlük tablosunun veri tabanı şeması

Çizelge 4.10. Oluşturulan sözlüğün ufak bir kısmı

Yaş Grubu	Kelime	Sıklık
Çocukluk	İlçe	27
Çocukluk	Öğretmen	101
Çocukluk	Okul	196
Ergenlik	Spor	82
Ergenlik	İlçe	30
Ergenlik	Takım	166
Yetişkinlik	İlçe	174
Yetişkinlik	Uyuşturucu	49
Yetişkinlik	Alkol	12

Oluşturulan sözlükteki kelimelerin yaş gruplarına göre sayısal dağılımının grafiği Şekil 4.6'da gösterilmektedir. Grafik incelendiğinde büyük oranda bir fark olmamakla beraber en fazla kelimenin barındığı yaş grubu Çocukluk olarak görülmektedir.



Şekil 4.6. Kelimelerin yaş gruplarına göre sayısal dağılımı

4.4. Tahminleme

Çalışmanın en önemli basamağı tahminlemedir. Veri setinin %30'luk kısmında bulunan haberler için bir yaş grubu tahmini yapılarak oluşturulan sözlük test edilmek istenmiştir. Sözlükte eğitim kısmında bulunan haberlerdeki seçili kelimeler (TF'i ortalamaya eşit ve büyük olanlar), etiketlenen yaş grubu bilgisi ve kelimenin yaş grubuna ait haberlerde ne kadar görüldüğü bilgisi (sıklık) bulunmaktadır.

Sınama haberlerine, eğitim haberlerine uygulanan veri ön işleme işlemleri yapıldıktan sonra her haber için bir yaş grubu tahmini yapılmak istenmiştir. Yaş grubu tahmini yapılırken sınama veri setindeki haberde bulunan her bir kelime için sözlükte var olan o kelimenin 3 yaş grubu içinde sıklığına bakılır. Habere ait her kelimenin sözlükteki her yaş grubuna uygun görülme sıklıklarına bakılarak bir puanlama işlemi yapılır (Çizelge 4.11). Haber hangi yaş grubu için en yüksek puanlamayı aldı ise test edilen haber o yaş grubu için uygun görülür.

Sınama için ayrılan haberler için de önceden uygun görülen bir (rehberlik ve psikolojik danışman tarafından manuel olarak belirlenen) yaş grubu bilgisi etiketlenmiştir. Böylece sınanan her haber için olması gereken (gerçek) ve tahmin edilen olmak üzere iki yaş grubu bilgisi yer almaktadır.

Çizelge 4.11. Yetişkinlik yaş grubu için tahmini puanlama aşaması

Yaş	Kuzey	Terör	Örgüt	Sevk	Pikap	Petrol	Devriye	Puan
1	35	0	12	2	0	7	0	56
2	11	1	4	1	0	1	1	19
3	122	415	370	94	1	31	31	**1064**

Çizelge 4.11'de örnek bir puanlama işlemi görülmektedir. Çizelge 4.11'de bulunan "kuzey", "terör", "örgüt", "sevk", "pikap", "petrol" ve "devriye" kelimeleri örnek olarak seçilen bir haberin veri ön işleme işleminden sonra elde edilen kelimeleridir. Daha sonra bu kelimelerin var olan sözlükte yaş grubuna uygun şekilde ne kadar görülme sıklığı kaydedildi ise yazılır. Çocukluk yaş grubunda; "kuzey" kelimesi 35 defa, "örgüt" kelimesi 12 defa, "sevk" kelimesi 2 defa, "petrol" kelimesi 7 defa görülmüştür ve "terör", "pikap", "devriye" kelimeleri hiç görülmemiştir. Toplamda bu haberin çocukluk yaş grubu için puanı 56'dır. Aynı işlemler diğer iki yaş grubu içinde yapılır. Sonuç olarak en yüksek skor 1064 ile Yetişkinlik grubunda görülür. Bu sebeple örnekteki haber Yetişkinlik yaş grubuna uygun olarak seçilir. Bu haberin ayrıca psikolojik danışman tarafından önceden etiketlenmiş yaş grubu (gerçek) Yetişkinlik olarak belirlenmiştir. Bu noktada bu haber için belirlenen gerçek yaş grubu ve puanlama sonrası tahmin edilen yaş grubu bilgisi birbirine uyuşmaktadır.

Çizelge 4.12. Çocukluk yaş grubu için tahmini puanlama aşaması

Yaş	Eğitim	Bakanlık	Ortaöğretim	Ortak	Sınav	Örnek	Soru	Tarih	Veli	Öğrenci	Okul	Sınıf	Bilgi	Puan
1	245	113	45	65	120	119	112	256	49	224	196	116	267	**1927**
2	41	14	0	20	7	28	52	86	1	25	23	8	66	371
3	55	163	1	75	15	35	81	112	2	38	33	22	171	803

Çizelge 4.12'de görülen örnekte ise yine bir başka haberin tahminlemesi yapılmıştır. Bu haberin "eğitim", "bakanlık", "ortaöğretim", "ortak", "sınav", "örnek", "soru", "tarih", "veli", "öğrenci", "okul", "sınıf", "bilgi" kelimeleri üç yaş grubu içinde sözlükteki sıklık bilgisine göre puanlanıp toplam puanın en fazla olduğu 1927 ile haberin Çocukluk yaş grubuna ait olduğu tahmini yapılmıştır. Bu haberin Gerçek yaş grubu da daha önceden Çocukluk olarak belirlenmiştir.

Çizelge 4.13. Ergenlik yaş grubu için tahmini puanlama aşaması

Yaş	**Oyun**	**Aile**	**Miras**	**Puan**
1	109	121	8	238
2	192	70	10	**272**
3	31	90	3	124

Son olarak Ergenlik yaş grubuna örnek bir haberde tahminleme işlemine bakılırsa Çizelge 4.13'de örnek bir haber bulunuyor. Haberin "aile", "miras", "oyun" kelimeleri bulunmaktadır. Bu kelimelerin sözlükteki yaş gruplarına uygun sıklıklarına göre en yüksek puanlaması Ergenlik yaş grubunda görülmektedir. Bu haberinde gerçek yaş grubu Ergenlik olarak belirlenmişti. Tüm haberler için bu puanlama işlemi tamamlandıktan sonra her haber için bir tahmin değeri ve bir gerçek değer elde edilerek bir listede tutulmaktadır.

5. SONUÇ VE ÖNERİLER

Sınamaya ayrılan haberler için haberin uygun görülen yaş grubu bilgisi (gerçek) önceden etiketlenmiş durumdadır. Tahminleme işlemi sonrası elde edilen tahmin değerinden sonra ise, son aşama olarak bu iki değerin birbiriyle karşılaştırılıp yapılan çalışmadaki sözlüğün doğruluğu üzerine çıkan sonuçların yorumlanmasıdır.

Bu çalışmada üç ayrı sözlük üzerine sonuçlar alınmıştır. Sözlük test sonuçlarını görmek için Hata Matrisi (Confusion Matrix) kullanılmıştır. Hata matrisi doldurulurken, "Gerçek" değeri için el ile atanmış yaş grubu bilgisi ve "Tahmin Edilen" değeri için ise tahminleme sonucu belirlenen yaş grubu bilgisi dikkate alınmıştır.

Birinci sözlükte veri ön işleme basamağında yer alan dil bilimi işlemlerinde (Morfoloji) metin etiketleme parçalarından fiiller hariç tüm kök tipleri (sıfat, isim, belirteç, zamir vb.) sözlüğe dahil edilmiştir.

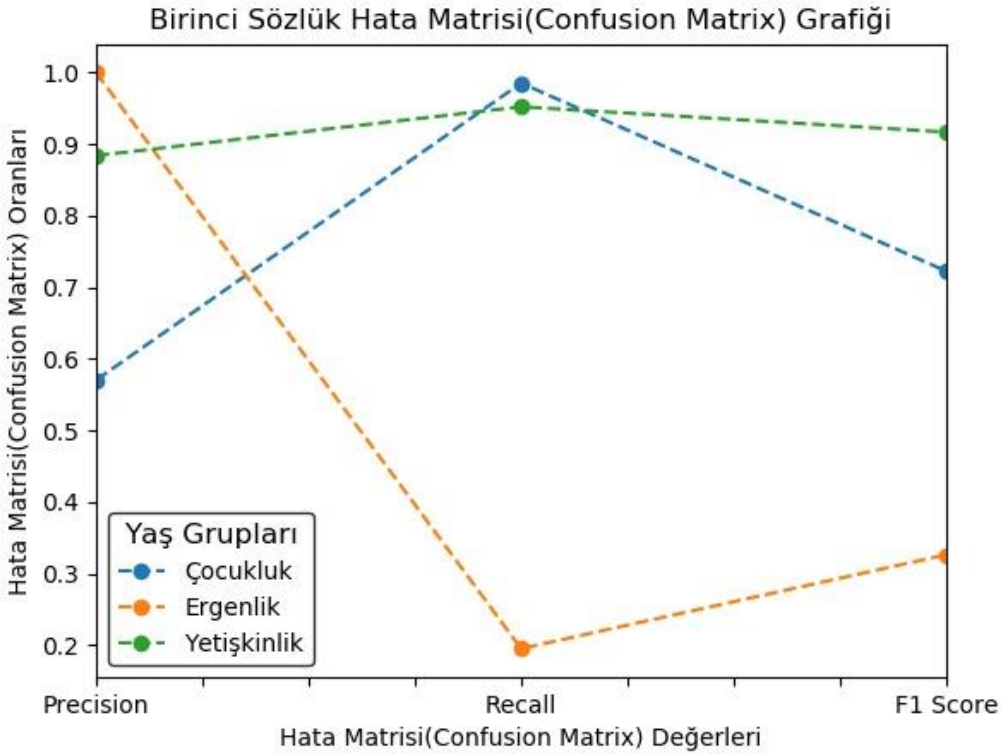
Sonuçlara göre sözlüğün Doğruluk (Accuracy) değeri %71 bulunmuştur. Çalışmanın Hata Matrisi değerleri Çizelge 5.1'de ve Hata Matrisinden üretilen sonuçlar ise Çizelge 5.2'de görülmektedir.

Çizelge 5.1. Birinci sözlüğün hata matrisi

		Tahmin Edilen		
		Çocukluk	**Ergenlik**	**Yetişkinlik**
Gerçek	**Çocukluk**	268	0	4
	Ergenlik	189	53	30
	Yetişkinlik	13	0	259

Çizelge 5.2. Birinci sözlüğün hata matrisinden elde edilen değerler

Yaş	Tutarlılık (Precision)	Hatırlama (Recall)	F1-Değeri
Çocukluk(6-13 yaş)	0.57	0.99	0.72
Ergenlik(13-18 yaş)	1.00	0.19	0.33
Yetişkinlik(18+ yaş)	0.88	0.95	0.92
Doğruluk			**0.71**
Ağırlıklı Ortalama	0.82	0.71	**0.66**



Şekil 5.1. Birinci sözlüğün Hata Matrisi grafiği

Çizelge 5.1'deki Hata Matrisinde görüldüğü gibi toplam 272 Çocukluk haberinden 268 tane haber başarılı bir şekilde doğru tahmin edilmiş olup 4 tanesi Yetişkinlik olarak tahmin edilmiştir. Çocukluk haberlerinden hiç biri Ergenlik yaş grubu için tahmin edilmemiştir. Aynı durum Yetişkinlik yaş grubunda da toplam 272 haberden 259 tanesi Yetişkinlik yaş grubu için başarılı bir şekilde tahmin edilmiş olup 13 tane haber ise Çocukluk yaş grubuna ait olduğuna dair tahminleme yapılmıştır. Yetişkinlik yaş grubu haberlerinden hiçbiri Ergenlik yaş grubu için uygun görülmemiştir. Son olarak Ergenlik yaş grubunda bulunan toplam 272 haberden sadece 53 tanesi Ergenlik için doğru

tahminleme yapılmıştır. 189 tane haber ile çoğunlukla Çocukluk yaş grubu için tahminleme yapılıp sonucunda Ergenlik yaş grubu için düşük bir tahminleme başarısı göstermiştir.

Çizelge 5.2'de yine bu hata matrisinden üretilen sonuçlarda da pozitif durumların ne kadar doğru tahmin edildiğini gösteren Hatırlama (Recall) sütununda Çocukluk ve Yetişkinlik için başarılı bir sonuç çıkarken Ergenlik yaş grubu için düşük bir sonuçla karşılaşılmaktadır. Tüm sınıflardan, doğru olarak ne kadar tahmin yapıldığının ölçüsü olan Tutarlılık (Precision) sütununda en başarılı sonuç Ergenlik yaş grubu için görülmektedir. Precision ve Recall sonuçlarının harmonik ortalaması olan F1-Değeri ise Yetişkinlik yaş grubunda en yüksek değere ulaşmıştır. Yine Şekil 5.1'de birinci sözlüğün Hata Matrisinden üretilen grafik üzerinde değerlerin dalgalanmaları görülmektedir.

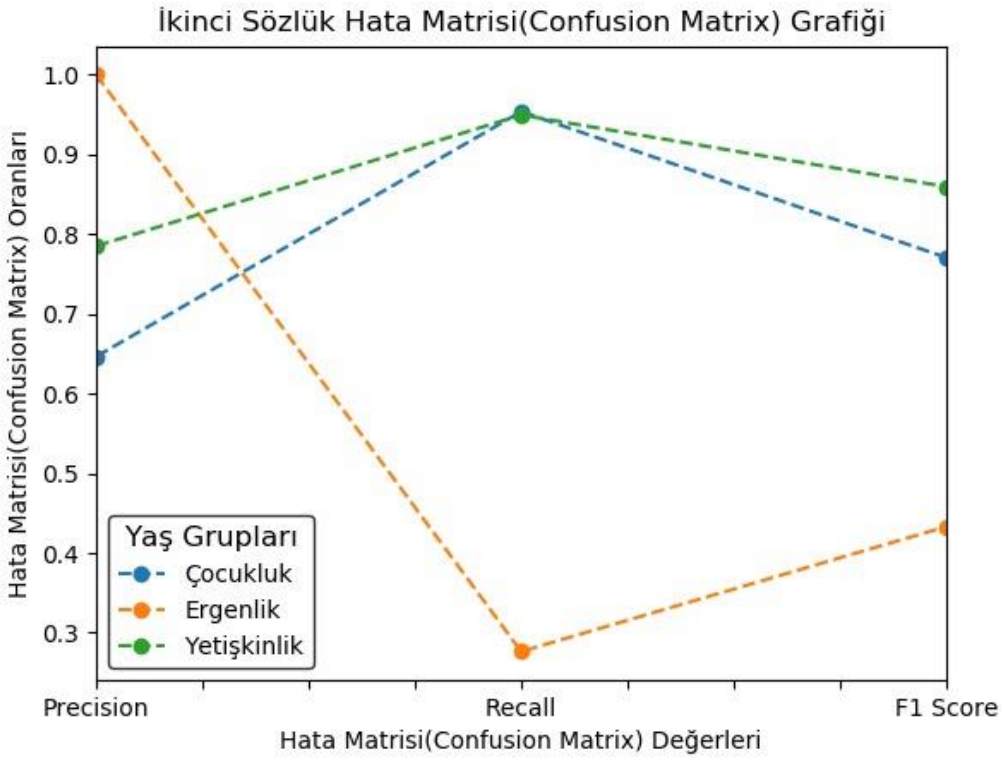
Gözlemler doğrultusunda sözlükte en seçici kelimelerin "isimler" olduğu görülüp birinci sözlükte fiil hariç tüm kök tiplerinin dahil edilmesinin aksine sadece "isim" olan kök tipleri sözlüğe dahil edilmiştir. Bu kısıtlama sonucu, sözlüğün başarısı %73 olarak kaydedilmiştir. Çalışmanın Hata Matrisi Çizelge 5.3'te ve Hata Matrisinden üretilen sonuçlar ise Çizelge 5.4'te gösterilmektedir.

Çizelge 5.3. İkinci sözlüğün hata matrisi

		Tahmin Edilen		
		Çocukluk	**Ergenlik**	**Yetişkinlik**
Gerçek	**Çocukluk**	309	0	15
	Ergenlik	155	81	57
	Yetişkinlik	14	0	263

Çizelge 5.4. İkinci sözlüğün hata matrisinden elde edilen değerler

Yaş	Precision	Recall	F1-Değeri
Çocukluk (6-13 yaş)	0.65	0.95	0.77
Ergenlik (13-18 yaş)	1.00	0.28	0.43
Yetişkinlik (18+ yaş)	0.79	0.95	0.86
Doğruluk			**0.73**
Ağırlıklı Ortalama	0.81	0.73	0.69



Şekil 5.2. İkinci sözlüğün Hata Matrisi grafiği

Çizelge 5.3'deki Hata Matrisinde görüldüğü gibi toplam 324 Çocukluk haberinden 309 tane haber başarılı bir şekilde tahmin edilmiş olup 15 tanesi Yetişkinlik olarak tahmin edilmiştir. Çocukluk haberlerinden hiç biri Ergenlik yaş grubu için tahmin edilmemiştir. Aynı durum Yetişkinlik yaş grubunda da toplam 277 haberden 263 tanesi Yetişkinlik yaş grubu için başarılı bir şekilde tahmin edilmiş ve 14 tane haber ise Çocukluk yaş grubuna ait olduğuna dair tahminleme yapılmıştır. Yetişkinlik yaş grubu haberlerinden hiçbiri Ergenlik yaş

grubu için uygun görülmemiştir. Son olarak Ergenlik yaş grubunda bulunan toplam 293 haberden sadece 81 tanesi Ergenlik için doğru tahminleme yapılmıştır. 155 tane haber ile çoğunlukla Çocukluk yaş grubu için tahminleme yapılıp Ergenlik yaş grubu için düşük bir tahminleme başarısı göstermiştir.

Çizelge 5.4'te yine bu Hata Matrisinden üretilen sonuçlarda da pozitif durumların ne kadar doğru tahmin edildiğini gösteren Recall sütununda Çocukluk ve Yetişkinlik için başarılı bir sonuç çıkarken Ergenlik yaş grubu için düşük bir sonuçla karşılaşılmaktadır. Tüm sınıflardan, doğru olarak ne kadar tahmin yapıldığının ölçüsü olan Precision sütununda en başarılı sonuç Ergenlik yaş grubu için görülmektedir. Precision ve Recall sonuçlarının harmonik ortalaması olan F1-Değeri ise Yetişkinlik yaş grubunda en yüksek değere ulaşmıştır. Yine Şekil 5.2'de birinci sözlüğün Hata Matrisinden üretilen grafik üzerinde değerlerin dalgalanmaları görülmektedir.

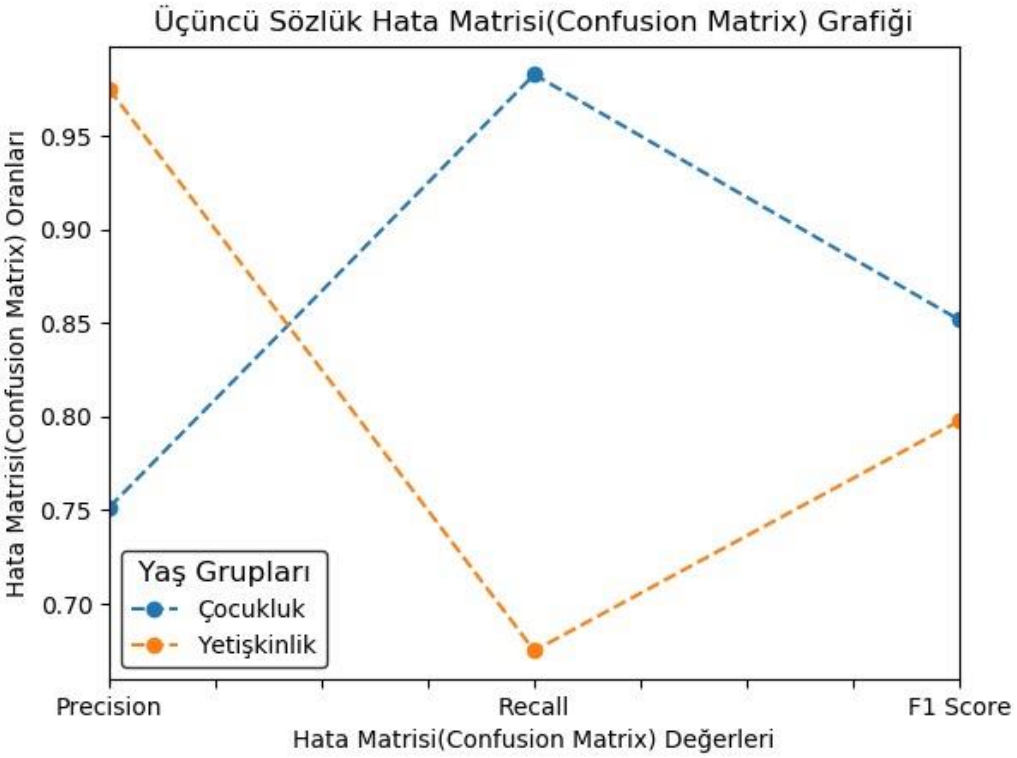
İkinci sözlükte sadece isimlerin olmasıyla birinci sözlüğe kıyasla başarı artırılmıştır. Ama iki sözlükte incelenip gözlemlendiğinde, Ergenlik yaş grubuna ait kelimelerin diğer her iki grupla örtüşmesinden dolayı sözlüğün başarısı düşmektedir. Ayrıca değerler incelendiğinde Ergenlik yaş grubu haberlerinin daha çok Çocukluk yaş grubuna ait olduğuna dair tahminlemeler yapılmıştır. Bu sebeple Ergenlik ve Çocukluk yaş grubu haberleri bir grup olmak üzere birleştirilerek veri tabanı güncellenmiştir. Sonucunda yaş grupları yetişkin (18+ yaş) ve yetişkin olmayan (6- 18 yaş) biçimde düzenlenmiştir. Bu şekilde sadece "isimlerin" bulunduğu üçüncü sözlükte ise %83 gibi daha ayrıştırıcı bir sonuca ulaşılmıştır. Çalışmanın Hata Matrisi Çizelge 5.5'te ve Hata Matrisinden üretilen sonuçlar ise Çizelge 5.6'da gösterilmektedir.

Çizelge 5.5. Üçüncü sözlüğün hata matrisi

		Tahmin Edilen	
		Çocukluk	**Yetişkinlik**
Gerçek	**Çocukluk (6-18 yaş)**	396	7
	Yetişkinlik (18+ yaş)	131	272

Çizelge 5.6. Üçüncü sözlüğün hata matrisinden elde edilen değerler

Yaş	Precision	Recall	F1-Değeri
Çocukluk (6-18 yaş)	0.75	0.98	0.85
Yetişkinlik (18+ yaş)	0.97	0.67	0.80
Doğruluk			**0.83**
Ağırlıklı Ortalama	0.86	0.83	**0.82**



Şekil 5.3. Üçüncü sözlüğün Hata Matrisi grafiği

Çizelge 5.5'teki Hata Matrisinde görüldüğü gibi toplam 403 Çocukluk yaş grubu haberinden 396 tane haber çok başarılı bir şekilde doğru tahmin edilmiş ve sadece 7 tanesi Yetişkinlik yaş grubu olarak yanlış tahmin edilmiştir. Yetişkinlik yaş grubunda da toplam 403 haberden 272 tanesi Yetişkinlik yaş grubu için Çocukluk yaş grubu haberlerine göre daha az başarılı bir şekilde tahmin edilmiştir. Yetişkinlik yaş grubunun 131 tane haberi ise Çocukluk yaş grubuna ait olduğuna dair yanlış tahminleme yapılmıştır.

Çizelge 5.6'da yine bu Hata Matrisinden üretilen sonuçlarda da pozitif durumların ne kadar doğru tahmin edildiğini gösteren Recall sütununda Çocukluk için başarılı bir sonuç çıkarken Yetişkinlik yaş grubu için daha düşük bir sonuçla karşılaşılmaktadır. Tüm sınıflardan, doğru olarak ne kadar tahmin yapıldığının ölçüsü olan Precision sütununda en başarılı sonuç Yetişkinlik yaş grubu için görülmektedir. Precision ve Recall sonuçlarının harmonik ortalaması olan F1-Değeri ise Çocukluk yaş grubunda en yüksek değere ulaşmıştır. Yine Şekil 5.3'te birinci sözlüğün Hata Matrisinden üretilen grafik üzerinde değerlerin dalgalanmaları görülmektedir.

Modelin görünmeyen veriler üzerinde ne kadar iyi performans göstereceğini öğrenmek, modelin doğruluğunu ölçmek için bir diğer yöntem olan k-fold çapraz doğrulamada kullanılarak üç sözlük test edilmiştir. K-fold çapraz doğrulama kullanılırken veri seti %80 eğitim %20 sınama için belirlenirken k değeri 5 olarak seçilmiştir ve elde edilen sonuçlar birinci, ikinci ve üçüncü sözlük için sırasıyla Şekil 5.4, Şekil 5.5 ve Şekil 5.6'da gösterilmektedir. Yaş grubu sütununda "1" Çocukluk yaş grubunu, "2" Ergenlik yaş grubunu ve "3" ise Yetişkinlik yaş grubunu temsil etmektedir.

Değer	K=1			K=2			K=3			K=4			K=5		
Yaş	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3
Precision	0.65	1.00	0.82	0.50	1.00	0.95	0.55	1.00	0.94	0.55	1.00	0.91	0.53	1.00	0.88
Recall	0.97	0.21	0.95	1.00	0.15	0.92	0.99	0.21	0.94	0.99	0.17	0.93	0.99	0.17	0.92
F1	0.78	0.35	0.88	0.66	0.26	0.93	0.70	0.35	0.94	0.71	0.29	0.92	0.69	0.29	0.90
Doğruluk	0.73			0.67			0.72			0.70			0.68		
Ortalama	0.70														

Şekil 5.4. Birinci sözlüğün k-fold değerleri

Değer	K=1			K=2			K=3			K=4			K=5		
Yaş	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3
Precision	0.70	1.00	0.77	0.54	1.00	0.86	0.60	0.98	0.88	0.60	1.00	0.84	0.58	0.98	0.85
Recall	0.97	0.29	0.95	0.98	0.23	0.92	0.98	0.30	0.94	0.95	0.26	0.95	0.97	0.24	0.95
F1	0.81	0.45	0.85	0.70	0.37	0.89	0.74	0.46	0.91	0.74	0.42	0.89	0.73	0.39	0.90
Doğruluk	0.75			0.69			0.75			0.73			0.72		
Ortalama	0.73														

Şekil 5.5. İkinci sözlüğün k-fold değerleri

Değer	K=1		K=2		K=3		K=4		K=5	
Yaş	1	3	1	3	1	3	1	3	1	3
Precision	0.86	0.96	0.82	0.96	0.79	0.96	0.79	0.99	0.83	0.99
Recall	0.98	0.69	0.99	0.60	0.99	0.56	1.00	0.57	1.00	0.64
F1	0.92	0.80	0.89	0.74	0.88	0.71	0.88	0.73	0.90	0.78
Doğruluk	0.89		0.85		0.83		0.83		0.87	
Ortalama	0.85									

Şekil 5.6. Üçüncü sözlüğün k-fold değerleri

K-katlamalı çapraz doğrulama sonuçları ile ilk elde edilen sonuçlar birbiri ile çok benzer değerlidir. İlk sözlük için %1'lik bir fark oluşurken üçüncü sözlük için %2'lik bir fark oluşmuştur. İkinci sözlükte değerler aynıdır.

Sonraki çalışmalarda geliştirilen yöntem için daha fazla haber toplanarak deneyin tekrarlanması düşünülmektedir. Ayrıca sözlük için anlamlı kelimelerin tespitinde kullanılan terim frekansı yöntemine alternatif olarak farklı yöntemler (Helmholtz ilkesi veya Rake algoritması) kullanılabilir. Bunun yanı sıra sözlük tabanlı bir model yerine, derin öğrenme modeli kullanılarak başarım değerlerinin kıyaslanması mümkündür.

KAYNAKLAR

Adalı, E., 2016. Doğal Dil İşleme. Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi, 5(2), 1-19.

Akgül E. S., Ertano C., Diri B., 2015. Twitter Verileri ile Duygu Analizi, Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi, 22(2), 106-110.

Akın, A.A., Akın, M.D., 2007. Zemberek, an Open Source NLP Framework for Turkish Languages, Structure, 10, 1-5.

Aktaş, Y., Yılmaz İnce, E., Çakır, A., 2017. Doğal Dil İşleme Kullanarak Bilgisayar Ağ Terimlerinin Wordnet Ontolojisinde Uyarlanması. Teknik Bilimler Dergisi, 7(2), 1-9.

Bertin, M., Atanassova, I., 2018. InTeReC: In-text Reference Corpus for Applying Natural Language Processing to Bibliometrics. 7th International Workshop on Bibliometric-enhanced Information Retrieval (BIR 2018) to be held as part of the 40th European Conference on Information Retrieval (ECIR), 26 March, Grenoble, France, 54-62.

Binici K., 2018. Kütüphane ve Bilgi Biliminde Tema ve Yönelim, Hiper yayın, 41s. İstanbul.

Bollegala, D., Alsuhaibani, M., Maehara, T., Kawarabayashi, K., 2016. Joint Word Representation Learning Using a Corpus and a Semantic Lexicon, AAAI.

Chiavetta, F., Bosco, G. L., Pilato, G., 2016. A Lexicon-based Approach for Sentiment Classification of Amazon Books Reviews in Italian Language. International Conference on Web Information Systems and Technologies, 23-25 April, Rome, 159-170.

Chowdhury, G. G., 2003, Natural Language Processing. Annual Review of Information Science and Technology, 37(1), 51–89.

Code Beautify, 2020. Erişim Tarihi: 16.03.2020
https://codebeautify.org/

Çok, F., 1993. Gelişim Psikolojisi, Kuramlar, Yöntemler ve Yaşamın İlk Yılları (kısaltarak çeviri). Ankara Üniversitesi Eğitim Bilimleri Fakültesi Dergisi, 2(26), 641-670.

Çekirdekci, S., Toptaş, V., Çekirdekçi, N., 2016. Bruner'in Zihinsel Gelişim İlkelerine Göre Yapılan Bilgisayar Destekli Eğitimin 3. Sınıf Geometri Kazanımlarının Başarı Ve Kalıcılığına Etkisi. Cumhuriyet Uluslararası Eğitim Dergisi, 5(5), 82-96.

Çelikkaya, G., 2015. Development of a Turkish Mobile Assistant Software Using Natural Language Processing Techniques, İstanbul Teknik Universty, 93p M.Sc. Thesis, İstanbul.

Demirel, M., Yörük, M., Özkan, O., 2013. Çocuklar İçin Güvenli İnternet: Güvenli İnternet Hizmeti ve Ebeveyn Görüşleri Üzerine Bir Araştırma - Safe Internet For Children: A Study on Safe Internet Service and Parental Views. Mehmet Akif Ersoy Üniversitesi Sosyal Bilimler Enstitüsü Dergisi, 4(7), 54-68.

Dinçer, B. T., 2004. Türkçe için İstatistiksel bir Bilgi Geri Getirim Sistemi, Ege Üniversitesi, Doktora Tezi, 407s, İzmir.

Durna, M., 2014. Identifying Event Nuggets in Turkish News Texts Using Natural Language Processing and Machıne Learning Methods, Boğaziçi University, M.Sc. Thesis, 69p, İstanbul.

Eryiğit, G., 2014. ITU Turkish Nlp Web Service, In Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL), Gothenburg, Sweden, April, Association for Computational Linguistics, 1-4.

Fırat, F., 2016. Çocuk Odak'sız' Habercilik: İnternet Gazetelerinde Çocuk İçerikli Haberlerin Sunumu ve Etik İhlaller, Gümüşhane Üniversitesi İletişim Fakültesi Elektronik Dergisi, 4(2), 1-17.

FrameNet, 2020. Erişim Tarihi: 18.03.2020
https://framenet.icsi.berkeley.edu/fndrupal/glossary

FrameNet Nedir, 2020. Erişim Tarihi: 18.03.2020
https://en.wikipedia.org/wiki/FrameNet#cite_note-Goddard2011-1

Gürses, İ., Kılavuz, M., 2011. Erikson'un Psiko-Sosyal Gelişim Dönemleri Teorisi Açısından Kuşaklararası Din Eğitimi ve İletişiminin Önemi. Uludağ Üniversitesi İlahiyat Fakültesi Dergisi, 20(2), 153-166.

Hata Matrisi, 2020. Erişim Tarihi: 21.03.2020
https://dev.to/overridedeveloper/understanding-the-confusion-matrix-264i

Hürriyet, 2020. Erişim Tarihi: 19.03.2020
https://www.hurriyet.com.tr/

Hürriyet RSS, 2020. Erişim Tarihi: 19.03.2020
http://dosyalar.hurriyet.com.tr/rss/

Işık, U., Koz, K., 2014. Çöp Yığınlarında Haber Aramak: İnternet Gazeteciliği Üzerine Bir Çalışma. Humanities Sciences, 9(2), 27-43.

İlhan, U., 2001. Application Of K-NN and FPTC Based Text Categorization Algorithms to Turkish News Reports.

K Katlamalı Çapraz Doğrulama, 2020. Erişim Tarihi: 19.05.2020
https://en.wikipedia.org/wiki/Cross-validation_(statistics)
https://medium.com/@tuncerergin/yapay-zekada-hold-out-cross-validation-nedir-1c6fae6de3a3
https://en.wikipedia.org/wiki/Cross-validation_(statistics)#/media/File:K-fold_cross_validation_EN.svg

Këpuska, V. Z., Rojanasthien, P., 2011. Speech Corpus Generation from DVDs of Movies and TV Series. Journal of International Technology and Information Management, 20(1), 49-82.

Khurana, D., Koli, A., Khatter, K., Singh, S., 2017. Natural Language Processing: State of The Art, Current Trends and Challenges.

Koeva, S., Stoyanova, I., Todorova, M., Leseva, S., 2016. Semi-automatic Compilation of the Dictionary of Bulgarian MultiwordExpressions. Proceedings of GLOBALEX 2016: Lexicographic Resources for Human Language Technology, Workshop at LREC2016, Portorož, Slovenia.

Kol, S., 2013. Erken Çocuklukta Bilişsel Gelişim ve Dil Gelişimi. Sakarya Üniversitesi Eğitim Fakültesi Dergisi, 21(21), 1-21.

Medium, 2020. Erişim Tarihi: 01.04.2020
https://medium.com/@ugrozkr_6539/zemberek-nlp-7add032881e9

Ofcom, 2020. Erişim Tarihi: 02.04.2020
https://www.ofcom.org.uk/research-and-data/media-literacy-research/childrens/children-and-parents-media-use-and-attitudes-report-2019

Oflazer, K., 2012. Türkçe ve Doğal Dil İşleme (Turkish Natural Language Processing). Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi, 2(5).

Ok, A., 2020. Erişim Tarihi: 16.03.2020
http://www.aliok.com.tr/projects/2014-10-02-trnltk.html

Özdemir, O., Özdemir, P., Kadak, M., Nasıroğlu, S., 2012. Kişilik Gelişimi. Psikiyatride Güncel Yaklaşımlar, 4(4), 566-589.

Patrick, H., James, P., 2005. A Pattern Dictionary for Natural Language Processing. Revue Française de Linguistique Appliquée, 2(X), 63-82.

Remove Stop Words, 2020. Erişim Tarihi: 05.04.2020
https://github.com/stopwords-iso/stopwords-tr
https://github.com/explosion/spaCy/blob/master/spacy/lang/tr/stop_words.py
https://github.com/ahmetaa/zemberek-nlp/blob/master/experiment/src/main/resources/stop-words.tr.txt

Pirim, H., 2006. Yapay Zeka. Journal of Yaşar University, 1(1) , 81-93.

Riloff, E., 1999. Automatically Constructing a Dictionary for Information Extraction Tasks. Proceedings of the Eleventh National Conference on Artificial Intelligence, AAAI Press / MIT Press, USA, 811–816.

Silverman, K. E., Anderson, V., Bellegarda, J.R, Lenzo, K.A., Naik, D., 1999. Design and Collection of a Corpus of Polyphones and Prosodic Contexts for Speech Synthesis Research and Development. 6th European Conference on Speech Communication and Technology, 5-9 September, Budapest, Hungary, 1-4.

Sucu, İ., Ataman, E., 2020. Dijital Evrenin Yeni Dünyası Olarak Yapay Zeka ve Her Filmi Üzerine Bir Çalışma. Yeni Medya Elektronik Dergisi, 4(1), 40-52.

The University of Chicago Press, 2020. Erişim Tarihi: 12.04.2020
https://press.uchicago.edu/ucp/books/book/chicago/E/bo3684144.html

Tsalidis, Ch., Vagelatos, A., Orphanos, G., 2004. An Electronic Dictionary as a Basis for NLP Tools: The Greek case, ArXiv cs.CL/0408061.

Turkish Natural Language Toolkit (TRNLTK), 2020. Erişim Tarihi: 16.03.2020
https://github.com/aliok/trnltk-java/blob/master/docs/102.md

Veri kümesi, 2020. Erişim Tarihi:01.06.2020,
https://github.com/rabiakontuk/HaberMetinlerininYasGruplari

Vijay, D., Bohra, A., Singh, V., Akhtar, S.S., Shrivastava, M., 2018. Corpus Creation and Emotion Prediction for Hindi-English Code-Mixed Social Media Text. Proceedings of NAACL-HLT 2018: Student Research Workshop, 2-4 June, New Orleans, Louisiana, 128-135.

WordNet, 2020. Erişim Tarihi: 27.03.2020
https://wordnet.princeton.edu/

Yapay Zeka, 2020. Erişim Tarihi: 03.04.2020
https://tr.wikipedia.org/wiki/Yapay_zek%C3%A2

Yumurtalı Ekmek, 2020. Erişim Tarihi: 18.03.2020
http://yumurtaliekmek.com/

Yumurtalı Ekmek RSS, 2020. Erişim Tarihi: 16.03.2020
http://yumurtaliekmek.com/feed/

Zemberek Kütüphanesi, 2020. Erişim Tarihi: 20.03.2020
https://github.com/ahmetaa/zemberek-nlp

EKLER

HaberRssTool.py

```
# -*- coding: utf-8 -*-
#Kütüphanelerimizi import ediyoruz.
from bs4 import BeautifulSoup as bs
from urllib.request import Request, urlopen
import feedparser as fp
import ssl
import sys
import string

## Veri tabanına ekleme/sorgulama işlemleri için ekliyoruz.
import VeritabaniIslemleri as vt

class HaberRssTool(object):
    """
    RSS servisinden Haberleri çözümleme işlerini yapan sınıf
    """
    def __init__(self,
title,link,imageUrl,details,description,categoryName,pubDate):
        self.Title = title
        self.Link = link
        self.ImageUrl = imageUrl
        self.Details = details
        self.Description = description
        self.CategoryName = categoryName
        self.PubDate = pubDate

    def HaberRssCozumleyici():

        ## Haberlerimizi çözümledikten sonra saklayacağımız değişken dizisi
        haberList = []

        ## linklerimizi alıyoruz.
        rssLink = HaberRssTool.HaberRssLinkList()

        ## rssLink listesinin içinden bir url seçiyoruz.
        for url in rssLink:

            ## Rss linklerini okuyup çözümleme işlemi
            rssXmlTags = fp.parse(url)

            ## RSS servisinin içindeki itemler Entries olarak geçiyor.
            print(f'Bulunan kayıt sayısı: {len(rssXmlTags.entries)} Rss Link: {url}')

            ## Entries içinden bir item seçiyoruz.
            for rssItem in rssXmlTags.entries:
```

```
            ## YumurtalıEkmek sitesi haberin detay bilgisinide RSS servisinde
sunmaktadır.
            if "yumurtaliekmek" in rssItem.link:
               haberItem = HaberRssTool(title = rssItem.title,
                           link = rssItem.link,
                        imageUrl = rssItem.media_content[0]["url"],
                        details = bs(rssItem.content[0].value,"html.parser").text,#
html taglerini temizleyip al
                     description = bs(rssItem.description,"html.parser").text,# html
taglerini temizleyip al
                     categoryName = rssItem.category,
                        pubDate = rssItem.published)
            else:
               ## Hürriyet haber sitesi haberin detay bilgisini RSS servisinde
sunmadığı için boş geçiyoruz.
               haberItem = HaberRssTool(title = rssItem.title,
                           link = rssItem.link,
                        imageUrl = rssItem.thumbnail["url"],
                        details = "",
                     description = bs(rssItem.description,"html.parser").text,# html
taglerini temizleyip al
                     categoryName = rssItem.category,
                        pubDate = rssItem.published)

            haberList.append(haberItem)

      ## Çözümlediğimiz haberleri saklamış olduğumuz değişkeni döndürüyoruz.
      return haberList

   def HaberRssLinkList():
      """
      http://dosyalar.hurriyet.com.tr/rss/ ve http://yumurtaliekmek.com/feed/
sitelerin RSS kısmından alındı.
      """
      rssLink = ["http://yumurtaliekmek.com/feed/",
               "http://www.hurriyet.com.tr/rss/anasayfa",
               "http://www.hurriyet.com.tr/rss/gundem",
               "http://www.hurriyet.com.tr/rss/ekonomi",
               "http://www.hurriyet.com.tr/rss/magazin",
               "http://www.hurriyet.com.tr/rss/spor",
               "http://www.hurriyet.com.tr/rss/dunya",
               "http://www.hurriyet.com.tr/rss/web-tv",
               "http://www.hurriyet.com.tr/rss/teknoloji",
               "http://www.hurriyet.com.tr/rss/saglik",
               "http://www.hurriyet.com.tr/rss/astroloji",
               "http://www.hurriyet.com.tr/rss/ankara",
               "http://www.hurriyet.com.tr/rss/ege"]
      return rssLink
```

```
  def HurriyetHaberDetayGetir(url):
    detay = ""
    try:
      #url içerisindeki html'i indiriyoruz.
      htmlRequest = urlopen(url = url, context =
ssl.SSLContext(ssl.PROTOCOL_SSLv23))
      htmlResponse = bs(htmlRequest,"html.parser")

      ##find_one kullandık çünkü sadece bir tane başlık var.
      haberBaslikBul = htmlResponse.find("h2", attrs = {"class":"rhd-article-
spot"})
      # Gelen html bilgisi içindeki h2 taglarının içinde class'ı "rhd-article-spot"
olan bilgiyi bul

      if haberBaslikBul == None:
        haberBaslikBul = htmlResponse.find("h1", attrs = {"class":"news-detail-
title"})
      if haberBaslikBul == None:##Video kategorisi için
        haberBaslikBul = htmlResponse.find("h1", attrs = {"class":"video-title"})
      if haberBaslikBul == None:##Mahmure kategorisi için
        haberBaslikBul = htmlResponse.find("h2", attrs = {"rhd-article-spot-
type-mahmure"})
      if haberBaslikBul == None: ##Seyahat kategorisi için
        h1VarMi = htmlResponse.find("h2", attrs = {"class":"news-spot hidden-
sm-down"})
        if h1VarMi != None:
          detay += htmlResponse.find("h1", attrs = {"class":"news-title"}).text
        haberBaslikBul = htmlResponse.find("h2", attrs = {"class":"news-spot
hidden-sm-down"})
      if haberBaslikBul == None:##eğitim kategorisi için
        haberBaslikBul = htmlResponse.find("h2", attrs = {"rhd-article-spot-
type-2"})

      if haberBaslikBul != None:
        haberBaslikText = haberBaslikBul.text
        detay += haberBaslikText

      haberDetayHtml = htmlResponse.find("div", attrs = {"class":"rhd-all-
article-detail"})
      # Gelen html bilgisi içindeki div'lerden class'ı "rhd-all-article-detail" olan
bilgiyi bul

      if haberDetayHtml != None:
        haberDetayPtagList = haberDetayHtml.find_all("p")
        for pTagItem in haberDetayPtagList:
          pTag = pTagItem.find("span",attrs={"style":"color: #0000ff;"}) #
Gelen P tagının içinde span tagı olup ve style bilgisi color olan bilgi varmı

          if pTag == None:
```

```
                detay +=" " + pTagItem.text
            else:
                break
        if haberDetayHtml == None:##burçlar için
            haberDetayHtml = htmlResponse.find("div", attrs = {"class":"horoscope-
detail-content"})
            if haberDetayHtml != None:
                haberDetayPtagList = haberDetayHtml.find_all("p")
                for pTagItem in haberDetayPtagList:
                    detay += pTagItem.text
        if haberDetayHtml == None:#eğitim kategorisi için
            haberDetayHtml = htmlResponse.find("div", attrs = {"class":"rhd-all-
article-detail-type-2"})
            if haberDetayHtml != None:
                haberDetayPtagList = haberDetayHtml.find_all("p")
                for pTagItem in haberDetayPtagList:
                    detay += " " + pTagItem.text
        if haberDetayHtml == None:
            haberDetayHtml = htmlResponse.find("div", attrs = {"class":"news-
detail-text"})
            if haberDetayHtml != None:
                haberDetayPtagList = haberDetayHtml.find_all("p")
                for pTagItem in haberDetayPtagList:
                    pTag = pTagItem.find("span",attrs={"style":"color: #0000ff;"}) #
Gelen P tagının içinde span tagı olup ve style bilgisi color olan bilgi var
                                                    # mı
                    if pTag == None:
                        detay += " " + pTagItem.text
                    else:
                        break
        if haberDetayHtml == None:
            haberDetayHtml = htmlResponse.find("div", attrs = {"class":"gallery-
main-container"})
            if haberDetayHtml != None:
                haberDetayHtml = htmlResponse.find_all("div", attrs =
{"class":"gallery-group"})

                for div in haberDetayHtml:
                    hText = div.find("h3",attrs={"class":"description"})
                    if hText != None:
                        detay +=" " + hText.text
        if haberDetayHtml == None:
            haberDetayHtml = htmlResponse.find("div", attrs = {"class":"video-
description"})
            if haberDetayHtml != None:
                detay +=" " + haberDetayHtml.text
        if haberDetayHtml == None:##lezizz kategorisi için
            haberDetayHtml = htmlResponse.find_all("div", attrs = {"class":"lz-
recipe-body"})
```

```
            if haberDetayHtml != None:
                for hbrDetay in haberDetayHtml:
                    haberDetayPtagList = hbrDetay.find_all("p")
                    for pTagItem in haberDetayPtagList:
                        detay += " " + pTagItem.text
            if haberDetayHtml == None:
                haberDetayHtml = htmlResponse.find_all("div", attrs = {"class":"news-
photo-content"})
                for div in haberDetayHtml:
                    hText = div.find("h2",attrs={"class":"description"})
                    if hText != None:
                        detay +=" " + hText.text

        except:
            detay = "Hata:{}".format(sys.exc_info())

        return detay

        def YumurtaliEkmekRssLinkList():
        """
        http://yumurtaliekmek.com/feed/ sitesinden alınmıştır.
        """
        rssLink = ["http://yumurtaliekmek.com/dunya/page/{sayfa}/",
                   "http://yumurtaliekmek.com/doga/page/{sayfa}/",
                   "http://yumurtaliekmek.com/egitim/page/{sayfa}/",
                   "http://yumurtaliekmek.com/etkinlik/page/{sayfa}/",
                   "http://yumurtaliekmek.com/genc-yazarlar/page/{sayfa}/",
                   "http://yumurtaliekmek.com/gezi-ve-tatil/page/{sayfa}/",
                   "http://yumurtaliekmek.com/hayvanlar/page/{sayfa}/",
                   "http://yumurtaliekmek.com/kitap/page/{sayfa}/",
                   "http://yumurtaliekmek.com/kisa-kisa/page/{sayfa}/",
                   "http://yumurtaliekmek.com/muzik/page/{sayfa}/",
                   "http://yumurtaliekmek.com/oyun/page/{sayfa}/",
                   "http://yumurtaliekmek.com/saglik/page/{sayfa}/",
                   "http://yumurtaliekmek.com/sanat/page/{sayfa}/",
                   "http://yumurtaliekmek.com/spor/page/{sayfa}/",
                   "http://yumurtaliekmek.com/tasarim/page/{sayfa}/",
                   "http://yumurtaliekmek.com/teknoloji-bilim/page/{sayfa}/",
                   "http://yumurtaliekmek.com/turkiye/page/{sayfa}/",
                   "http://yumurtaliekmek.com/tv-sinema/page/{sayfa}/",
                   "http://yumurtaliekmek.com/yeme-icme/page/{sayfa}/"]
        return rssLink

    def YumurtaliEkmekRssBilgiAl(url, dbBaglanti):

        haberList = []

        sayfaNo = 0
        isPaging = True
```

```
    while isPaging:

        sayfaNo += 1
        tempLink = url

        sayfaLink = tempLink.replace("{sayfa}",str(sayfaNo))
        headers = {'User-Agent': 'Mozilla/5.0'}
        request = Request(sayfaLink, headers=headers)

        try:
            htmlRequest = urlopen(request,context =
ssl.SSLContext(ssl.PROTOCOL_SSLv23)).read()
            htmlResponse = bs(htmlRequest,"html.parser")

            haberContent = htmlResponse.find("div", attrs = {"class":"wrap
container"})

            haberCategoryBul = haberContent.find("div", attrs = {"class":"page-
header span8 single-content"}).find("h1").text
            categoryName = haberCategoryBul.translate({ord(c): None for c in
string.whitespace})

            haberDetayHeaderList = haberContent.find_all("header", attrs =
{"class":"item"})
            for header in haberDetayHeaderList:

                haberDescription = header.find("p").text

                haberLinkHtml = header.find("a", attrs = {"class":"itemTitle"})
                haberLink = haberLinkHtml.attrs['href']

                if vt.VeritabaniIslemleri.HaberlerKayitSorgula(dbBaglanti,
haberLink)[0] >= 1:
                    continue

                haberTitle = haberLinkHtml.text

                haberImageHtml = header.find("img", attrs = {"class":"wp-post-
image"})
                haberImage = haberImageHtml.attrs['src']
                obj =
HaberRssTool(haberTitle,haberLink,haberImage,"",haberDescription,categoryN
ame,"")
                haberList.append(obj)

        except Exception as hata:
            isPaging = False
```

```
    return haberList
  def YumurtaliEkmekDetayGetir(url):

    pubDate = ""
    haberTitle = ""
    haberDetayText = ""

    headers = {'User-Agent': 'Mozilla/5.0'}
    request = Request(url, headers=headers)

    try:
      #url içerisindeki html'i indiriyoruz.
      htmlRequest = urlopen(request,context =
ssl.SSLContext(ssl.PROTOCOL_SSLv23)).read()
      htmlResponse = bs(htmlRequest,"html.parser")

      pubDate = htmlResponse.find("time", attrs =
{"class":"updated"}).attrs['datetime']
      haberTitle = htmlResponse.find("h1", attrs = {"class":"entry-title"}).text

      #Title bilgisini haber içerikten silme
      titleHtml = htmlResponse.find("h1", attrs = {"class":"entry-title"})
      titleHtml.decompose()

      #haber içerik
      haberContent = htmlResponse.find("div", attrs = {"class":"entry-
content"})

      #related_post divlerini haber içerikten silme
      relatedPostDivs = haberContent.find_all("div", attrs =
{"class":"related_post"})
      for x in relatedPostDivs:
        haberContent.select_one("div.related_post").decompose()

      #clearfix divlerini haber içerikten silme
      clearfixDivs = haberContent.find_all("div", attrs = {"class":"clearfix"})
      for x in clearfixDivs:
        haberContent.select_one("div.clearfix").decompose()

      #script taglerini haber içerikten silme
      scriptTags = haberContent.find_all("script")
      for x in scriptTags:
        haberContent.script.decompose()

      haberDetayText = haberContent.text

    except:
      hataVarMi = True
      haberDetayText = "Hata:{}".format(sys.exc_info())
```

```
        returnModel =
HaberRssTool(haberTitle,url,"","",haberDetayText,"","",pubDate)

        return returnModel

    def YumurtaliEkmekHaberRssBilgiCekKaydet():

        haberLinkListesi = []

        haberLinkListesi = HaberRssTool.YumurtaliEkmekRssLinkList()

        ##Baglanti açma işlemini yapıyoruz.
        dbBaglanti = vt.VeritabaniIslemleri.BaglantiAc(vt.DATABASE_PATH)

        ## Haber Rss linklerini sırayla dönüyoruz.
        for link in haberLinkListesi:

            haberList = HaberRssTool.YumurtaliEkmekRssBilgiAl(link, dbBaglanti)
            print(f"{link} - Count : {len(haberList)}")

            ## Çektiğimiz haberList'in içinde her haber için
Title,Description,CategoryName,Link vs bilgiler var Bu bilgileri db'ye kaydetmek
için her haberi sırasıyla dönüyoruz.

            for haberModel in haberList:
                try:
                    haberTitle = haberModel.Title
                    haberLink = haberModel.Link
                    haberCategoryName = haberModel.CategoryName
                    haberDescription = haberModel.Description
                    haberImageUrl = haberModel.ImageUrl
                    ## Haber'e ait linkten haberin içeriğini almaya gidiyoruz.
                    detay = HaberRssTool.YumurtaliEkmekDetayGetir(haberModel.Link)
                    haberPubDate = detay.PubDate
                    haberDetails = detay.Details
                    ## Sonra ilgili haber'e ait toplamış olduğumuz bilgileri kaydediyoruz.
                    haberObj =
HaberRssTool(haberTitle,haberLink,haberImageUrl,haberDetails,haberDescript
ion,haberCategoryName,haberPubDate)
                    vt.VeritabaniIslemleri.HaberlerKayitEkle(dbBaglanti, haberObj)
                except Exception as hata:
                    print(hata)

        ## Bağlantımızı kapatıyoruz.
        vt.VeritabaniIslemleri.BaglantiKapat(dbBaglanti)

if __name__ == "__main__":
```

```
    ## RSS'in içerisindeki xml'leri model haline getirme.
    haberList = HaberRssTool.HaberRssCozumleyici()

    ## Baglanti açma işlemini yapıyoruz.
    dbBaglanti = vt.VeritabaniIslemleri.BaglantiAc(vt.DATABASE_PATH)

    ## Modellenen haberleri sırasıyla dönme
    for haberModel in haberList:
        ## Kaydetme işlemini yap.
        vt.VeritabaniIslemleri.HaberlerKayitEkle(dbBaglanti,haberModel)

    ## Daha önceden kaydedilmiş fakat detayı boş olan haberleri güncelleme
işlemi
    detayBosHaberList =
vt.VeritabaniIslemleri.HaberlerDetayiBosOlanlar(dbBaglanti)
    print("Sorgulanan toplam kayıt sayısı:{}".format(len(detayBosHaberList)))
    for haberDetay in detayBosHaberList:
        haberLink = haberDetay[3]

        haber_detay = ""
        if "hurriyet" in haberLink:
            haber_detay = HaberRssTool.HurriyetHaberDetayGetir(haberLink)
        else:
            haber_detay = HaberRssTool.YumurtaliEkmekDetayGetir(haberLink)

        if haber_detay == "":
            continue
        else:

vt.VeritabaniIslemleri.HaberlerKayitGuncelle(dbBaglanti,haber_detay,haberLin
k)

    ##Kayıt işlemleri tamamladıktan sonra db bağlantısı kapatıyorum.
    vt.VeritabaniIslemleri.BaglantiKapat(dbBaglanti)

    ## Ekstra YumurtaliEkmek.com sitesi için yazılmıştır.  Haberleri çeker
kaydeder.
    HaberRssTool.YumurtaliEkmekHaberRssBilgiCekKaydet()
```

VeritabaniIslemleri.py

```
# -*- coding: utf-8 -*-
import os
import sqlite3 as sq
from sqlite3 import Error
from datetime import datetime # date(tarih) tip dönüşümü için kullanıyoruz.
from bdateutil import parser  # date(tarih) tip dönüşümü için kullanıyoruz.

# Veritabanımızın bulunduğu dosya yolu.
```

```
DATABASE_PATH : str = os.path.join(os.getcwd(),
"Data","Veritabani","HaberSqlLiteDb.db")
DATABASE_PATH_1_SCORE : str = os.path.join(os.getcwd(),
"Data","Veritabani","HaberSqlLiteDb_1_Score.db")
DATABASE_PATH_2_SCORE : str = os.path.join(os.getcwd(),
"Data","Veritabani","HaberSqlLiteDb_2_Score.db")
DATABASE_PATH_3_SCORE : str = os.path.join(os.getcwd(),
"Data","Veritabani","HaberSqlLiteDb_3_Score.db")
class VeritabaniIslemleri(object):
    """
    Veritabanı oluşturma, tablo oluşturma, kayıt ekleme/güncelleme/listeleme
işlemlerini yapar.
    """
    def BaglantiAc(dbPath):
        """ SQLite veritabanına veritabanı bağlantısı oluşturma
        :return: Bağlantı nesnesi veya None(Yok) bilgisi dönme
        """
        baglanti = None

        try:
            ## Veritabanı yoksa aynı zamanda oluşturma işlemini de yapmaktadır.
            baglanti = sq.connect(dbPath)
        except Error as hata:
            print(f"Veritabanı oluşturma ya da bağlanma işlemi yapılırken hata
oluştu. Detay: {hata}")

        return baglanti

    def BaglantiKapat(baglanti):
        """
        Açık olan veri tabanı bağlantısını kapatır.
        """
        if (baglanti): # Açık bağlantı varsa kapat
            baglanti.close()


    def TabloOlustur(baglanti, sqlText):

        try:
            komut = baglanti.cursor()
            komut.execute(sqlText)
            komut.close()
        except Error as hata:
            print(f"Veritabanına tablo oluşturma işlemi yapılırken hata oluştu. Detay:
{hata}")

    def VeriTabaniOlusturma(dbPath, sozlukTabloOlusturma,
haberTabloOlusturma):
        """
```

```
    Veri tabanı oluşturma işlemlerini yapar.
    Veri tabanı silindiğinde çalıştırılır.
    """

    # Baglanti Açma ve Db oluşturma
    baglanti = VeritabaniIslemleri.BaglantiAc(dbPath)


    if sozlukTabloOlusturma == "E":
        # Sozluk tablosunu oluşturan sql kodu
        SOZLUK_CREATE_SQL_TEXT = "CREATE TABLE IF NOT EXISTS Sozluk
(kelime text NOT NULL,sıklık integer NOT NULL,grup integer NOT NULL);"
        # Sözlük isimli tablo oluşturma işlemi
        VeritabaniIslemleri.TabloOlustur(baglanti,SOZLUK_CREATE_SQL_TEXT)


    if haberTabloOlusturma == "E":
        # Haberler tablosunu oluşturan sql kodu
        HABERLER_CREATE_SQL_TEXT = "CREATE TABLE IF NOT EXISTS
Haberler (Title TEXT,Description TEXT,Details TEXT,Link TEXT NULL,MediaUrl
TEXT,CategoryName TEXT,PubDate NUMERIC,InsertDate NUMERIC);"
        # Haberler isimli tablo oluşturma işlemi

VeritabaniIslemleri.TabloOlustur(baglanti,HABERLER_CREATE_SQL_TEXT)

    # İşlemler tamamlandıktan sonra bağlantıyı kapatıyoruz.
    VeritabaniIslemleri.BaglantiKapat(baglanti)


  def SozlukTabloSil(baglanti):
    """
    Sozluk tablosunun içeriğini komple siler.
    """
    try:
      komut = baglanti.cursor()
      komut.execute(f"delete from Sozluk")
      baglanti.commit()
    except Error as hata:
      print(f"Sozluk tablosundan kayıtlar silinemedi. Detay: {hata}")

  def SozlukKelimeEkle(baglanti, kelime, sıklık, grup):
    """
    İlgili parametlere göre Sozluk tablosuna kayıt ekleme işlemini yapar.
    """
    try:
      komut = baglanti.cursor()
      komut.execute(f"INSERT INTO [Sozluk] ([kelime],[sıklık],[grup])VALUES
('{kelime}',{sıklık},{grup})")
      baglanti.commit()
```

```
        except Error as hata:
            print(f"Sozluk tablosuna kayıt ekleme işlemi yapılırken hata oluştu.
Detay: {hata}")

    def SozlukTopluKelimeEkle(sozlukDictKelimeList, dbPath):

        baglanti = VeritabaniIslemleri.BaglantiAc(dbPath)

        VeritabaniIslemleri.SozlukTabloSil(baglanti)

        ## Sözlüğe kelimeleri ekleme işlemi
        for index, key in sozlukDictKelimeList.items():
            kelime = key.get("kelime")
            sıklık = key.get("sıklık")
            grup = key.get("grup")
            VeritabaniIslemleri.SozlukKelimeEkle(baglanti,kelime,sıklık,grup)

        ## İşlemler tamamlandıktan sonra bağlantıyı kapatıyoruz.
        VeritabaniIslemleri.BaglantiKapat(baglanti)

    def SozlukKelimeHesapla(baglanti, kelimeList):

        kelimeWhere = "','".join(map(str,kelimeList))

        sqlKomut = f"select grup, sum(sıklık) as skor from [sozluk] where kelime in
('{kelimeWhere}') group by grup"
        try:
            ## açmış olduğumuz bağlantıyı cursor nesnesine tanımlıyoruz.
            komut = baglanti.cursor()
            komut.execute(sqlKomut)
            sonuc = komut.fetchall()
            return sonuc
        except Error as hata:
            print(f"Sozluk tablosunda işlem yapılırken hata oluştu. Detay: {hata}")

    def HaberlerKayitSorgula(baglanti, haberLink):
        """
        'haberLink' parametresine göre Haberler tablosunda ilgili linke ait
haberlerin sayısını döner.
        """
        sqlKomut = "SELECT Count(*) FROM [Haberler] Where
Link='{}'".format(haberLink)
        try:
            komut = baglanti.cursor()
            komut.execute(sqlKomut)
            sonuc = komut.fetchone()
            return sonuc
        except Error as hata:
```

```
        print(f"Haber tablosunda sorgulama işlemi yapılırken hata oluştu. Detay:
{hata}")

    def HaberlerKayitlariGetir(baglanti, adet):
        """
        Verilan adet değişkenine ait haber getirir.
        """
        sqlKomut = "SELECT TOP {} * FROM [Haberler]".format(adet)
        try:
            komut = baglanti.cursor()
            komut.execute(sqlKomut)
            sonuc = komut.fetchall()
            return sonuc
        except Error as hata:
            print(f'Haber tablosunda işlem yapılırken hata oluştu. Detay: {hata}')

    def HaberlerKayitGuncelle(baglanti, haberDetay, haberLink):
        try:
            komut = baglanti.cursor()
            komut.executemany('UPDATE Haberler SET [Details] = ? WHERE [Link] =
?',[(haberDetay,haberLink)])
            baglanti.commit()
        except Error as hata:
            print(f"Haber tablosu güncelleme işlemi yapılırken hata oluştu. Detay:
{hata}")

    def HaberlerDetayiBosOlanlar(baglanti):

        sqlKomut = "select * from Haberler h Where h.Details = ''"
        try:
            komut = baglanti.cursor()
            komut.execute(sqlKomut)
            sonuc = komut.fetchall()
            return sonuc
        except Error as hata:
            print(f'Haber tablosunda işlem yapılırken hata oluştu. Detay: {hata}')

    def HaberlerKayitEkle(baglanti, HaberObj):
        """
        Verilen HaberObj nesnesine göre Haberler tablosuna kayıt ekleme işlemi
yapar.
        """
        try:

            haberLink = HaberObj.Link
            ## haberLink'e ait daha önce eklenmiş haber var ise ekleme işlemini
yapma
            if VeritabaniIslemleri.HaberlerKayitSorgula(baglanti, haberLink)[0] >= 1:
                pass
```

```
        else:
                now = datetime.now()
            # format bilgimizi dd/mm/YY H:M:S
            dt_string = now.strftime("%d/%m/%Y %H:%M:%S")

            komut = baglanti.cursor()
            komut.executemany("INSERT INTO [Haberler]
([Title],[Description],[Details],[Link],[MediaUrl],[CategoryName],[PubDate],[Ins
ertDate])VALUES
(?,?,?,?,?,?,?,?)",[(HaberObj.Title,HaberObj.Description,HaberObj.Details,HaberOb
j.Link,HaberObj.ImageUrl,HaberObj.CategoryName,parser.parse(HaberObj.PubD
ate),parser.parse(dt_string))])
            baglanti.commit()
        except Error as hata:
            print(f"Haber tablosuna ekleme işlem yapılırken hata oluştu. Detay:
{hata}")

if __name__ == "__main__":

    VeritabaniIslemleri.VeriTabaniOlusturma(DATABASE_PATH,"H","E")

VeritabaniIslemleri.VeriTabaniOlusturma(DATABASE_PATH_1_SCORE,"E","H")

VeritabaniIslemleri.VeriTabaniOlusturma(DATABASE_PATH_2_SCORE,"E","H")

VeritabaniIslemleri.VeriTabaniOlusturma(DATABASE_PATH_3_SCORE,"E","H")
    print("Veri tabanı oluşturma işlemleri tamamlandı.")
```

ZemberekTool.py

```
import re
import ssl
import math
from collections import Counter
import pandas as pd


##Zemberek.jar dosyasını açmak için kullandığım kütüphaneler
from jpype import JClass, getDefaultJVMPath, isJVMStarted, java, shutdownJVM,
startJVM, JString
from typing import List

#StopWords dosyalarını indirmek için kullandığım kütüphaneler
from nltk import download
from nltk.corpus import stopwords

##Stopwords dosyamı import ediyorum.
import MyStopwords as stp
```

```
## Veri tabanına ekleme işlemleri için kullanılan kütüphaneler
from VeritabaniIslemleri import VeritabaniIslemleri as vt

import os
UYGULAMA_PATH = os.getcwd()
ZEMBEREK_PATH : str = os.path.join(UYGULAMA_PATH,
'Zemberek','0.17.1','zemberek-full.jar')

EXCEL_PATH : str =
os.path.join(UYGULAMA_PATH,"Data","HaberVeriSeti_Skor_1_2.xlsx")
EXCEL_PATH_3_SCORE : str =
os.path.join(UYGULAMA_PATH,"Data","HaberVeriSeti_Skor_3.xlsx")

SOZLUK_PATH_1_SCORE : str =
os.path.join(UYGULAMA_PATH,"Data","SozlukResult","HaberSozluk_1_Skor.xlsx
")
SOZLUK_PATH_2_SCORE : str =
os.path.join(UYGULAMA_PATH,"Data","SozlukResult","HaberSozluk_2_Skor.xlsx
")
SOZLUK_PATH_3_SCORE : str =
os.path.join(UYGULAMA_PATH,"Data","SozlukResult","HaberSozluk_3_Skor.xlsx
")

if isJVMStarted() == False:
   startJVM(getDefaultJVMPath(),'-ea',f'-
Djava.class.path={ZEMBEREK_PATH}',convertStrings=False)

TurkishTokenizer : JClass = JClass('zemberek.tokenization.TurkishTokenizer')
Token : JClass = JClass('zemberek.tokenization.Token')
Type : JClass = JClass('zemberek.tokenization.Token.Type')
TurkishMorphology : JClass =
JClass('zemberek.morphology.TurkishMorphology')
SentenceAnalysis : JClass =
JClass('zemberek.morphology.analysis.SentenceAnalysis')
SingleAnalysis : JClass = JClass('zemberek.morphology.analysis.SingleAnalysis')
TurkishSpellChecker : JClass =
JClass('zemberek.normalization.TurkishSpellChecker')
RootLexicon : JClass = JClass('zemberek.morphology.lexicon.RootLexicon')
AnalysisFormatters : JClass =
JClass('zemberek.morphology.analysis.AnalysisFormatters')
WordAnalysis : JClass = JClass('zemberek.morphology.analysis.WordAnalysis')

StopWordList =
stp.STOP_WORDS.union(stp.AYLAR).union(stp.GUNLER).union(stp.ILLER)
StopWordsDict = Counter(StopWordList)

## Kelimenin hangi dile ait olduğunu tespit eder.
LanguageIdentifier : JClass = JClass('zemberek.langid.LanguageIdentifier')
```

```
lid = LanguageIdentifier.fromInternalModelGroup("tr_group")

Morphology : TurkishMorphology = (TurkishMorphology.builder().
                        setLexicon(RootLexicon.getDefault())
                        #.ignoreDiacriticsInAnalysis()
                        .useInformalAnalysis()
                        .build())
## Kelimenin doğru yazılışı için öneride bulunur.
SpellChecker : TurkishSpellChecker = TurkishSpellChecker(Morphology)

class ZemberekTool:


    @classmethod
    def TFHesapla(cls, kelimeList):
        counts = Counter(kelimeList)

        return dict(counts)

    @classmethod
    def TokenizationSentence(cls, sentence):
        """
        Verilen cümleye tokenization işlemi uygular.
        """
        sentence = re.sub("-", " ", sentence)
        tokenList = java.util.ArrayList()
        tokenizer : TurkishTokenizer =
TurkishTokenizer.builder().ignoreTypes(Token.Type.SpaceTab,
            Token.Type.NewLine,
            Token.Type.Punctuation,
            Token.Type.RomanNumeral,
            Token.Type.Number,
            Token.Type.PercentNumeral,
            Token.Type.Time,
            Token.Type.Date,
            Token.Type.URL,
            Token.Type.Email,
            Token.Type.HashTag,
            Token.Type.Mention,
            Token.Type.MetaTag,
            Token.Type.Emoji,
            Token.Type.Emoticon,
            Token.Type.UnknownWord,
            Token.Type.Unknown).build()

        for token in tokenizer.tokenizeToStrings(sentence):
            tokenList.add(token)

        return tokenList
```

```
  @classmethod
  def MorphologySentence(cls, sentence, skor):
    """
    Verilen cümledeki kelimelerin köklerini bulup geri döndürür.
    """
    resultList : java.util.ArrayList = java.util.ArrayList()

    if len(sentence) == 0:
      return sentence
    else:
      analysis : SentenceAnalysis =
Morphology.analyzeAndDisambiguate(JString(sentence))

      for e in analysis:

        word = str(e.getWordAnalysis().getInput())

        ## En iyi kökü "best" değişkenine atıyoruz.
        best : SingleAnalysis = e.getBestAnalysis()

        ## Kelime'nin sıfat mı fiil mi isim mi bağlaç mı olduğunu tespit ettiğimiz
değişken.
        #wordPos = f'{best.getPos()}'
        wordShortPos = f'{best.getPos().shortForm}'

        if best.isUnknown():## Kelime tanımsız ve herhangi bir kök işlemi yok
ise ekleme işlemini yapma
          continue
        else:
          ## Kelimeye ait kelime kökünü "lemmas" değişkenine atıyorum.
          lemmas : java.util.ArrayList = best.getLemmas()
          lemmasStr = f'{str(lemmas[0])}'
          ## Kelimenin pos'u bu listeden biriyse sözlüğe ekleme.
          ## Conjunction (bağlaç), Verb (Fiil), Numeral (Numara), Determiner
(Belirteç), Question (soru mi mu misiniz)
          ## Pronoun (zamir), PostPositive (edat), Duplicator (tekrarlanan
kıpır kıpır, cıvıl cıvıl), Adverb (zarf)
          ## Interjection (ünlem), Adjective (Sıfat)
          if skor == 1:
            kelimePosList = [ "Verb"]
          else:
            kelimePosList = ["Conj", "Verb", "Adj", "Num", "Det", "Ques", "Pron",
"Postp", "Dup", "Adv","Interj"]

          ## Kelime'nin POS'u yukarıdaki listenin birisi ise es geç
          if wordShortPos in kelimePosList:
            continue
          else:
            ##Bulunan kelimenin köküde isim ise ekle değilse ekleme
```

```
                analysis2 : SentenceAnalysis =
Morphology.analyzeAndDisambiguate(JString(lemmasStr))
                for e in analysis2:
                    word2 = str(e.getWordAnalysis().getInput())
                    best2 : SingleAnalysis = e.getBestAnalysis()
                    wordShortPos2 = f"{best2.getPos().shortForm}"

                    if wordShortPos2 != wordShortPos:
                        continue
                    else:
                        if skor == 1:
                            resultList.add(lemmasStr)
                        else:
                            # Bulunan kelime türkçe mi ? kontrolü
                            languageControl = str(lid.identify(word2))
                            if languageControl != "tr":
                                ## Bulunan kelime türkçe değilse es geç
                                continue
                            else:
                                ## Bulunan kelime türkçe ise listeye ekle
                                resultList.add(lemmasStr)

        return resultList

    @classmethod
    def RemoveStopWords(cls, wordList):
        """
        Verilen kelime listesindeki stop word'leri kaldırır.
        """
        filteredWordListStr = " ".join([str(word) for word in wordList if word not in
StopWordsDict])
        filteredWordList = filteredWordListStr.split(" ")

        return filteredWordList

    @classmethod
    def JvmStop(cls):
        """JVM'yi kapatır."""
        if isJVMStarted() == True:
            shutdownJVM()

    @classmethod
    def PrepareHaberData(cls, corpusSet, skor):
        """
        Verilen corpusu işleyerek hazırlanmış data olarak geri döner. (train ya da
test hangisi verilirse)
        """
        train_prepared_corpus = []
        for index, row in corpusSet.iterrows(): ## satır satır datayı okuyorum.
```

```
            print("İşlenen haber:",len(train_prepared_corpus))
            yas = row["Yaş"]
            details = row["Details"]
            #print("Haber Detay: ", details, '\n')

            tokenization = cls.TokenizationSentence(details)
            tokenization_processed = " ".join([str(token) for token in tokenization])
            #print("Tokenization: ", tokenization_processed, '\n')

            morphology = cls.MorphologySentence(tokenization_processed, skor)
            morphology_processed = " ".join([str(token) for token in morphology])
            #print("Morphology: ", morphology_processed, '\n')

            ## Haber'e ait isim olan kelime yoksa devam et
            if len(morphology) == 0:
              continue

            removeStopWords = cls.RemoveStopWords(morphology)
            removeStopWords_processed = " ".join([str(text) for text in
removeStopWords])
            #print("RemoveStopWords: ", removeStopWords_processed, '\n')

            tf = cls.TFHesapla(removeStopWords)
            #print("TF: ",tf,'\n')

            model = dict(haber_detay = details,
                    haber_yas_tip = yas,
                    haber_kelime_list = removeStopWords,
                    haber_kelime_tf_list = tf)

            train_prepared_corpus.append(model)

        return train_prepared_corpus
    @classmethod
    def KelimeListOlustur(cls, preparedCorpus):
        kelimeList = {}
        IslemYapilanHaberSayi = 0

        for haber_model in preparedCorpus:
            IslemYapilanHaberSayi = IslemYapilanHaberSayi + 1
            print("Sözlük İşlenmiş Haber:",IslemYapilanHaberSayi)

            ## Haber'e ait yaşTip, KelimeList ve KelimeFrekansList bilgilerini
alıyoruz.
            haberYasTip = haber_model.get("haber_yas_tip")
            haberKelimeList = haber_model.get("haber_kelime_list")
            haberKelimeTFList = haber_model.get("haber_kelime_tf_list")

            ## Ortalama sıklık bilgisini bulmmak için
```

```
        haberKelimeTFListSum = sum(haberKelimeTFList.values())
        haberToplamKelimeSayisi = len(haberKelimeList)
        haberKelimeOrtalama = haberKelimeTFListSum /
haberToplamKelimeSayisi

        ## Habere ait kelimeleri sırasıyla dön
        for kelime, sıklık in haberKelimeTFList.items():

          kelimeKriter = f"{kelime}|{haberYasTip}"


          if sıklık < haberKelimeOrtalama:
            continue ## Ortalamanın altında kaldığı için listeye eklemeyip bir
sonraki kelime geç
          else:
            # İlgili kelime sözlük'e daha önce eklenmiş mi?
            if kelimeKriter in kelimeList:
              ## Daha önce eklenmiş ise kelime'ye ait "sıklık" bilgisi alınır.
              secilenKelimeSıklık = kelimeList[kelimeKriter].get("sıklık")
              ## Daha önce eklenen kelimenin grubu aynı olduğu için "sıklık"
değerini 1 arttırıyoruz.
              kelimeList[kelimeKriter].update(sıklık = secilenKelimeSıklık + 1)
            else:
              ## Daha önce eklenmemişse sözlüğe ekle
              sozlukEkleModel = dict(kelime = kelime, sıklık = 1, grup =
haberYasTip)
              kelimeList[kelimeKriter] = sozlukEkleModel

      return kelimeList

    @classmethod
    def KelimeListesiniExceleAktarma(cls, kelimeList, excelPath):
      """
      Verilen kelime listesini verilen hedef klasör yoluna kaydeder
      """
      ## "kelimeList" isimli listemizi DataFrame haline getiriyoruz
      df = pd.DataFrame(kelimeList)
      df = df.T


      excelWriterModel = pd.ExcelWriter(excelPath, engine ='xlsxwriter')

      df.to_excel(excelWriterModel, sheet_name ='Sözlük')

      ## Bütün eklemeler yapıldıktan sonra save() fonksiyonuyla dosyamızı
kaydedip kapatıyoruz.
      excelWriterModel.save()

    @classmethod
```

```
  def HaberTahminiYapma(cls, preparedCorpus, dbPath):
    """
    Evulate işlemleri yapar,
    Geriye actual ve predicted bilgisi döner
    """
    actual = []
    predicted = []

    ## Hazırlanmış corpus bilgisini sırayla dön
    for haber_model in preparedCorpus:

      etiketlenenHaberYasTip = haber_model.get("haber_yas_tip")
      haberKelimeTFList = haber_model.get("haber_kelime_tf_list")

      ## Veri tabanı bağlantı açma işlemi
      dbBaglanti = vt.BaglantiAc(dbPath)

      ## haberKelimeTFList listesindeki kelimelerin yaş grupları karşılığını
hesaplama işlemi
      testSonuc =  vt.SozlukKelimeHesapla(dbBaglanti, haberKelimeTFList)

      ## Değişkenlerimizi tanımladık
      grup_2_skor = 0
      grup_3_skor = 0
      grup_4_skor = 0

      ## testSonuc kısmından dönen kayıtlarımızı 2 , 3 ve 4 yas grubumuza
göre sonuçları kontrol ediyoruz
      for row in testSonuc:
        rowGrup = row[0] ## Yaş grubu
        rowSkor = row[1] ## Frekans toplamı

        if rowGrup == 2: ## 2 nolu gruba ait ise skoru ekle
          grup_2_skor = grup_2_skor + rowSkor
        elif rowGrup == 3: ## 3 nolu gruba ait ise skoru ekle
          grup_3_skor = grup_3_skor + rowSkor
        elif rowGrup == 4: ## 4 nolu gruba ait ise skoru ekle
          grup_4_skor = grup_4_skor + rowSkor
        else: ## degilse es geç (Hiç bir kelime tanımlı değilse)
          continue

      ## Sonuçları ekrana yazdırma
      print("*"*15)
      print(f"Grup 2 Skor: ", grup_2_skor)
      print(f"Grup 3 Skor: ", grup_3_skor)
      print(f"Grup 4 Skor: ", grup_4_skor)

      ## En büyük skoru olan yaş grubunu bul
      tahminGrup = 0
```

```
            if (grup_2_skor >= grup_3_skor) and (grup_2_skor >= grup_4_skor):
                tahminGrup = 2
            elif (grup_3_skor >= grup_2_skor) and (grup_3_skor >= grup_4_skor):
                tahminGrup = 3
            else:
                tahminGrup = 4

            print("İşlenen Haber Yas tip:",str(etiketlenenHaberYasTip), "Hesaplanan
", grup_2_skor,",",grup_3_skor,"ve",grup_4_skor," skorları içinde büyük olan
tahminin yaş grubu: ",tahminGrup)
            actual.append(etiketlenenHaberYasTip)
            predicted.append(tahminGrup)

        return actual, predicted

class nltk_download:
    def __init__(self):
        try:
            _create_unverified_https_context = ssl._create_unverified_context
        except AttributeError:
            pass
        else:
            ssl._create_default_https_context = _create_unverified_https_context
        download()
```

TezProject.py

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-

## Programda kullanacağımız kütüphanelerimizi tanımlıyoruz.
import pandas as pd
import ZemberekTool as ztool
from ZemberekTool import ZemberekTool as zt
import VeritabaniIslemleri as vt

## İşlemlerin karışmaması adına programın eğitilmesini istiyorsak "islem"
değişkenine "Train" ya da test edilmesini istiyorsak "Test" yazmamız
gerekmektedir.
## Train , Test
islem = "Test"

## Çalışmada 3 farklı sonuç elde edilmiştir. Hangi sonucu görmek istiyorsak
onu yazmalıyız.
## 1, 2, 3
skor = 3
if skor == 1:
    P_EXCEL_PATH = ztool.EXCEL_PATH
    P_SOZLUK_PATH = ztool.SOZLUK_PATH_1_SCORE
```

```
    P_VERITABANI_PATH = vt.DATABASE_PATH_1_SCORE
if skor == 2:
    P_EXCEL_PATH = ztool.EXCEL_PATH
    P_SOZLUK_PATH = ztool.SOZLUK_PATH_2_SCORE
    P_VERITABANI_PATH = vt.DATABASE_PATH_2_SCORE
if skor == 3:
    P_EXCEL_PATH = ztool.EXCEL_PATH_3_SCORE
    P_SOZLUK_PATH = ztool.SOZLUK_PATH_3_SCORE
    P_VERITABANI_PATH = vt.DATABASE_PATH_3_SCORE

data = pd.read_excel(io = P_EXCEL_PATH, sheet_name = "Haberler")

data = data.drop_duplicates()

df = pd.DataFrame(data,  columns = ["Details","Yaş"])

print("Veri seti hakkında bilgi")
print(df.info(),"\n")

print("Veri setindeki yaş gruplarına göre toplam haber sayısı:")
print(df['Yaş'].value_counts(),"\n")

## Veri setinde detayı boş olan kayıtları kaldırdık.
df = df.dropna()

## skor 1 ve 2 olan dataseti'nden eşit sayıda haber alınması için eklendi.
if skor == 1 or skor == 2:
    ## Veri setindeki her yaştan 1000 tane haber alma
    df_2_nolu_haberler = df[df['Yaş'] == 2].iloc[0:1000]
    df_3_nolu_haberler = df[df['Yaş'] == 3].iloc[0:1000]
    df_4_nolu_haberler = df[df['Yaş'] == 4].iloc[0:1000]
    ## Veri setini birleştirme
    df = pd.concat([df_2_nolu_haberler, df_3_nolu_haberler, df_4_nolu_haberler])

## Veri setimizi Eğitim ve Test olarak bölme | sklearn kütüphanesi
from sklearn.model_selection import train_test_split
trainingSet, testSet = train_test_split(df, test_size = 0.3, random_state = 0)

print("Veri seti bölündükten sonraki toplamları hakkında bilgi")
print(f"Toplam Data Boyutu: {len(df)}")
print(f"Train Data Boyutu: {len(trainingSet)}")
print(f"Test Data Boyutu: {len(testSet)}","\n")

## Train işlemleri
if islem == "Train":
    print("Train veri setindeki ilgili haberlere ait yaş adet bilgileri:")
    print(trainingSet['Yaş'].value_counts())

    ## Data işleme ve corpus oluşturma işlemi.
```

```
    ## Tokenization, Morphology, Remove Stop Words aşamalarından geçirilmiş
düzenlenmiş datayı attığımız değişken
    train_prepared_corpus = zt.PrepareHaberData(trainingSet, skor)

    ## Hazırlanmış corpus için kelime listesi oluşturma
    kelimeList = zt.KelimeListOlustur(train_prepared_corpus)

    ## Oluşan "kelimeList"'esini excele kaydetme (Analiz yapmak için
kullanıyoruz.)
    ## Normal şartlarda veri tabanına eklenmektedir.
    sozlukExcelKaydet = zt.KelimeListesiniExceleAktarma(kelimeList,
P_SOZLUK_PATH)

    ## Oluşan "kelimeList"'esini veri tabanına kaydetme
    sozlukDbKaydet = vt.VeritabaniIslemleri.SozlukTopluKelimeEkle(kelimeList,
P_VERITABANI_PATH)

    ## train işlemlerini yaptıktan sonra test işlemlerine geçmesi için değişkeni
güncelliyoruz.
    islem = "Test"

## Test işlemleri
if islem == "Test":
    print("Test veri setindeki ilgili haberlere ait yaş adet bilgileri:")
    print(testSet['Yaş'].value_counts())

    ## skor 1 datasını test ederken eşit sayıda haber almak için eklendi.
    if skor == 1:
        ## Veri setindeki her yaş grubundan eşit sayıda haber alma işlemi.
        test_df_2_nolu_haberler = testSet[testSet['Yaş'] == 2].iloc[0:272]
        test_df_3_nolu_haberler = testSet[testSet['Yaş'] == 3].iloc[0:272]
        test_df_4_nolu_haberler = testSet[testSet['Yaş'] == 4].iloc[0:272]
        ## Veri setini birleştirme
        testSet = pd.concat([test_df_2_nolu_haberler, test_df_3_nolu_haberler,
test_df_4_nolu_haberler])

    ## Data işleme ve corpus oluşturma işlemi.

    test_prepared_corpus = zt.PrepareHaberData(testSet, skor)

    evulate = zt.HaberTahminiYapma(test_prepared_corpus,
P_VERITABANI_PATH)
    actual = evulate[0] ## actual listesi alınır.
    predicted = evulate[1] ## predicted listesi alınır.

    ## confusion matrix oluşturma işlemleri.
    from sklearn.metrics import confusion_matrix
    from sklearn.metrics import accuracy_score
    from sklearn.metrics import classification_report
```

```
    print("Confusion Matrix :","\n",confusion_matrix(actual, predicted) )
    print("Accuracy Score :", accuracy_score(actual, predicted))
    print("Report :","\n",classification_report(actual, predicted))

print("İşlemler tamamlandı.")
## İşlemler tamamlandıktan sonra Zemberek bağlantısını kapatıyoruz.
zt.JvmStop()
print("Zemberek kapatıldı.")
```

ÖZGEÇMİŞ

Adı Soyadı : Rabia KONTUK

Doğum Yeri ve Yılı : İstanbul, 10/01/1995

Medeni Hali : Bekar

Yabancı Dili : İngilizce

E-posta : rabiakontuk@gmail.com

Eğitim Durumu

Lise : Sabiha Gökçen Anadolu Kız Teknik Lisesi, 2013

Lisans : Kocaeli Üniversitesi, Eğitim Fakültesi, BÖTE Bölümü, 2017

Yüksek Lisans :İstanbul Ticaret Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Anabilim Dalı, 2020

Mesleki Deneyim

Canada Schools,
Bilişim Teknolojileri Öğretmeni 2018-...(devam ediyor)

Yayınları

Kontuk, R., Turan, M.. 2020. NLP Kullanılarak Haberlerin Yaş Gruplarına Göre Sınıflandırılması. Gazi Üniversitesi Fen Bilimleri Dergisi Part C: Tasarım ve Teknoloji, 8(2), 372-382.