KOCAELİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ
ANABİLİM DALI

YÜKSEK LİSANS TEZİ

TÜRKÇE HABER METİNLERİNİN YAKLAŞIK EN YAKIN KOMŞU ARAMA YÖNTEMİ İLE BENZERLİK TESPİTİ

HAKAN TORUN

KOCAELİ 2022

KOCAELİ ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ

ANABİLİM DALI

YÜKSEK LİSANS TEZİ

TÜRKÇE HABER METİNLERİNİN YAKLAŞIK EN YAKIN KOMŞU ARAMA YÖNTEMİ İLE BENZERLİK TESPİTİ

HAKAN TORUN

Dr. Öğr. Üyesi Alpaslan Burak İNNER		
Danışman,	**Kocaeli Üniv.**	**.....**
Prof.Dr. Sevinç İLHAN OMURCA		
Jüri Üyesi,	**Kocaeli Üniv.**	**.....**
Dr. Öğr. Üyesi Burcu YILMAZ		
Jüri Üyesi,	**Gebze Teknik Üniv.**	**.....**

Tezin Savunulduğu Tarih: 13.05.2022

ETİK BEYAN VE ARAŞTIRMA FONU DESTEĞİ

Desteği Kocaeli Üniversitesi Fen Bilimleri Enstitüsü tez yazım kurallarına uygun olarak hazırladığım bu tez/proje çalışmasında,

- Bu tezin/projenin bana ait, özgün bir çalışma olduğunu,
- Çalışmamın hazırlık, veri toplama, analiz ve bilgilerin sunumu olmak üzere tüm aşamalarında bilimsel etik ilke ve kurallara uygun davrandığımı,
- Bu çalışma kapsamında elde edilen tüm veri ve bilgiler için kaynak gösterdiğimi ve bu kaynaklara kaynakçada yer verdiğimi,
- Bu çalışmanın Kocaeli Üniversitesi'nin abone olduğu intihal yazılım programı kullanılarak Fen Bilimleri Enstitüsü'nün belirlemiş olduğu ölçütlere uygun olduğunu,
- Kullanılan verilerde herhangi bir tahrifat yapmadığımı,
- Tezin/Projenin herhangi bir bölümünü bu üniversite veya başka bir üniversitede başka bir tez/proje çalışması olarak sunmadığımı, beyan ederim.

☒ Bu tez/proje çalışmasının herhangi bir aşaması hiçbir kurum/kuruluş tarafından maddi/altyapı desteği ile desteklenmemiştir.

Herhangi bir zamanda, çalışmamla ilgili yaptığım bu beyana aykırı bir durumun saptanması durumunda, ortaya çıkacak tüm ahlaki ve hukuki sonuçları kabul ettiğimi bildiririm.

Hakan TORUN

YAYIMLAMA VE FİKRİ MÜLKİYET HAKLARI

Fen Bilimleri Enstitüsü tarafından onaylanan lisansüstü tezimin/projemin tamamını veya herhangi bir kısmını, basılı ve elektronik formatta arşivleme ve aşağıda belirtilen koşullarla kullanıma açma izninin Kocaeli Üniversitesi'ne verdiğimi beyan ederim. Bu izinle Üniversiteye verilen kullanım hakları dışındaki tüm fikri mülkiyet haklarım bende kalacak, tezimin/projemin tamamının ya da bir bölümünün gelecekteki çalışmalarda (makale, kitap, lisans ve patent vb.) kullanımı bana ait olacaktır.

Tezin/projenin kendi özgün çalışmam olduğunu, başkalarının haklarını ihlal etmediğimi ve tezimin/projenin tek yetkili sahibi olduğumu beyan ve taahhüt ederim. Tezimde yer alan telif hakkı bulunan ve sahiplerinden yazılı izin alınarak kullanılması zorunlu metinlerin yazılı izin alarak kullandığımı ve istenildiğinde suretlerini Üniversiteye teslim etmeyi taahhüt ederim.

Yükseköğretim kurulu tarafından yayınlanan ***"Lisansüstü Tezlerin Elektronik Ortamda Toplanması, Düzenlenmesi ve Erişime Açılmasına İlişkin Yönerge"*** kapsamında tezim aşağıda belirtilen koşullar haricinde YÖK Ulusal Tez Merkezi/ Kocaeli Üniversitesi Kütüphaneleri Açık Erişim Sisteminde erişime açılır.

- ☐ Enstitü yönetim kurulu kararı ile tezimin/projemin erişime açılması mezuniyet tarihinden itibaren 2 yıl ertelenmiştir.
- ☐ Enstitü yönetim kurulu gerekçeli kararı ile tezimin/projemin erişime açılması mezuniyet tarihinden itibaren 6 ay ertelenmiştir.
- ☒ Tezim/projem ile ilgili gizlilik kararı verilmemiştir.

Hakan TORUN

ÖNSÖZ VE TEŞEKKÜR

Bu tez çalışması kapsamında Türkçe yayın yapan haber sitelerinde yayınlanan haberlerin benzerlik tespitinin yapılması, farklı metin vektörleştirme yöntemlerinin performanslarını ve metin özetlemenin benzerlik tespiti sürecindeki etkisini incelemek amacıyla uzun soluklu bir çalışma gerçekleştirilmiştir. Haberciliğin modern dünyadaki yansıması ve gelişmekte olan ülkelerdeki haberciliğin durumu maalesef pek iç açıcı değildir. Bu çalışmada bilgisayar bilimlerinde önemli bir problem olan en yakın komşu arama problemi haber metinlerinin benzerlik tespiti için kullanılmış ve Türkçe haber yayınları arasındaki aldatıcı haberlerin geldiği korkutucu duruma dikkat çekilmeye çalışılmıştır.

Çalışmalarım süresince bana destek veren kıymetli hocam Dr. Öğr. Üyesi Burak İNNER'e teşekkürlerimi sunarım.

Mayıs – 2022 Hakan TORUN

İÇİNDEKİLER

ETİK BEYAN VE ARAŞTIRMA FONU DESTEĞİ i
YAYIMLAMA VE FİKRİ MÜLKİYET HAKLARI ii
ÖNSÖZ VE TEŞEKKÜR iii
İÇİNDEKİLER iv
ŞEKİLLER DİZİNİ vi
TABLOLAR DİZİNİ vii
SİMGELER VE KISALTMALAR DİZİNİ viii
ÖZET xi
ABSTRACT x
1. GİRİŞ 1
2. MAKİNE ÖĞRENMESİ 6
2.1. Makine Öğrenmesi Yöntemleri 7
2.1.1. Gözetimli Öğrenme 7
2.1.2. Gözetimsiz Öğrenme 8
2.1.3. Yarı Gözetimli Öğrenme 8
2.1.4. Pekiştirmeli Öğrenme 9
2.2. Yapay Sinir Ağları 9
2.3. K-en Yakın Komşu Algoritması 15
3. YAKLAŞIK EN YAKIN KOMŞU ARAMA 17
3.1. Mesafe Ölçütleri 19
3.1.1. Öklid Mesafesi 19
3.1.2. Manhattan Mesafesi 20
3.1.3. Minkowski Mesafesi 20
3.1.4. Kosinüs Benzerliği 21
3.1.5. Hamming Mesafesi 22
3.2. Yaklaşık En Yakın Komşu Arama Kütüphaneleri 23
3.2.1. ANNOY 23
3.2.2. FALCONN 23
3.2.3. FLANN 24
3.2.4. Hnswlib 24
4. METİN ÖZETLEME SİSTEMLERİ 25
4.1. Çıkarıma Dayalı Özetleme 25
4.1.1. Textrank Algoritması 26
4.1.2. Lexrank Algoritması 27
4.1.3. Luhn Algoritması 28
4.1.4. Latent Semantic Analysis (LSA) 30
4.2. Yoruma Dayalı Özetleme 31
4.3. Özetleme Performans Değerlendirme Metrikleri 32
4.3.1. Rouge 32
4.3.2. Bleu 35
5. DİL MODELİ GELİŞTİRME 36
5.1. Bag of Words (BOW) 36
5.2. TF-IDF 37
5.3. N-Gram 38
5.4. Word2Vec 39
5.5. Doc2Vec 40

5.6. fastText 42
5.7. GloVe 42
6. MALZEME VE YÖNTEM 44
6.1. Veri Toplama 44
6.2. Ön İşleme 46
6.3. Otomatik Metin Özetleyici 47
6.4. Haber Vektörlerinin Oluşturulması 48
6.5. Vektör İndekslerinin Oluşturulması 50
6.6. Benzerlik Tespiti 51
6.7. Performans Analizi 52
7. BULGULAR VE TARTIŞMA 55
8. SONUÇLAR VE ÖNERİLER 60
KAYNAKLAR 62
KİŞİSEL YAYIN VE ESERLER 66
ÖZGEÇMİŞ 67

ŞEKİLLER DİZİNİ

Şekil 1.1.	Sistem Tasarımı ……..……....	3
Şekil 2.1.	Biyolojik sinir hücresinin temsili gösterimi ……..	10
Şekil 2.2.	Yapay sinir ağı gösterimi ……...	11
Şekil 2.3.	Bazı aktivasyon fonksiyonları ……...	12
Şekil 2.4.	Transfer fonksiyonu …….……..	12
Şekil 2.5.	Tek katmanlı yapay sinir ağı modeli ……..	13
Şekil 2.6.	OR kapısının YSA ile modellenmesi …….	13
Şekil 2.7.	Çok katmanlı yapay sinir ağı modeli …….	14
Şekil 2.8.	K-en yakın komşu algoritması eğitim verisi örnek gösterim …..…….	16
Şekil 3.1.	K-en yakın komşu algoritmasının tipik bir görselleştirmesi …...……..	17
Şekil 3.2	Yaklaşık en yakın komşu aramanın vektör uzayında gösterimi ……	18
Şekil 3.3.	Oklid mesafesinin gösterimi …….…….	19
Şekil 3.4.	Manhattan mesafesinin gösterimi …….…….	20
Şekil 3.5.	Kosinüs benzerliğinin gösterimi …….……...	21
Şekil 3.6.	Küp üzerinde Hamming mesafesi ……..	22
Şekil 4.1.	TextRank algoritması akış diyagramı ……	27
Şekil 4.2.	Luhn algoritmasında cümlenin önem faktörünün hesaplanması ……...	29
Şekil 5.1.	Word2Vec mimarisi ……...	39
Şekil 5.2.	Kelime temsil vektörlerinin oluşturulması ……	40
Şekil 5.3.	Paragraf vektörlerinin oluşturulması ……	41
Şekil 6.1.	Veri toplayıcı sisteme dair akış ……..	44
Şekil 6.2.	On işleme aracına dair akış ……	46
Şekil 6.3.	Haber vektörlerinin oluşturulmasına dair akış ……...	48
Şekil 6.4.	Vektör indekslerinin oluşturulmasına dair akış …….	50
Şekil 6.5.	Benzer haberlerin tespit edilmesi sürecine dair akış ……..	51
Şekil 6.6.	Uzman oylaması için geliştirilen web tabanlı sistem - 1 …...……	53
Şekil 6.7.	Uzman oylaması için geliştirilen web tabanlı sistem - 2 …...……	53

TABLOLAR DİZİNİ

Tablo 2.1. Bazı toplayıcı fonksiyonlar 11
Tablo 3.1. Iki karakter dizisi arasındaki Hamming mesafeleri 22
Tablo 4.1. Ornek bir doküman listesi 30
Tablo 4.2. Terim-Doküman matrisi 31
Tablo 4.3. Makine özeti ve referans özet 33
Tablo 5.1. Kelime sıklık tablosu 36
Tablo 5.2. Bag of Words modeli 36
Tablo 5.3. Ornek cümlelerden TF dönüşümü 37
Tablo 5.4. Ornek cümlelerden IDF dönüşümü 38
Tablo 5.5. Ornek cümlelerden TF-IDF dönüşümü 38
Tablo 6.1. Toplanan veri istatistiği 45
Tablo 6.2. Eğitim verisine dair veri istatistiği 45
Tablo 6.3. Türkçe durak kelimelerden örnekler 47
Tablo 6.4. Otomatik metin özetleyici ile özetlenen metin-1 47
Tablo 6.5. Otomatik metin özetleyici ile özetlenen metin-2 48
Tablo 6.7. Ornek TF-IDF matrisi 49
Tablo 6.8. Ornek Doc2Vec vektör matrisi 49
Tablo 6.9. Tespit edilen benzer haber örnekleri-1 52
Tablo 6.10. Tespit edilen benzer haber örnekleri-2 52
Tablo 7.1. Benzerlik tespiti mesafe ölçütlerine göre performans karşılaştırması ... 55
Tablo 7.2. Vektörleştirme yöntemlerine göre sistem başarısı 58

SİMGELER VE KISALTMALAR DİZİNİ

r : Recall (Kesinlik)
p : Precision (Hassasiyet)
Q : Sorgu Noktası
v : Vektör Boyutu

Kısaltmalar

ANN : Approximate Nearest Neighbors (Yaklaşık En Yakın Komşular)
BOW : Bag of Words (Kelime Çantası)
CBOW : Continuous Bag of Words (Sürekli Kelime Çantası)
DB : Database (Veritabanı)
GloVe : Global Vectors (Global Vektörler)
HTML : Hypertext Markup Language (Hiper Metin Işaretleme Dili)
IDF : Inverse Document Frequency (Ters Doküman Frekansı)
k-NN : k-Nearest Neighbors (K-en Yakın Komşular)
LCS : Longest Common Subsequence (En Uzun Ortak Küme)
LSA : Latent Semantic Analysis (Gizli Anlam Analizi)
LSI : Latent Semantic Indexing (Gizli Semantik Indeksleme)
LSH : Locality-Sensitive Hashing (Yerel-Duyarlı Hashleme)
NNS : Nearest Neighbor Search (En Yakın Komşu Arama)
NSW : Navigable Small World (Gezinilebilir Küçük Dünya)
PR : PageRank (Sayfa Sıralaması)
RSS : Rich Site Summary (Zengin Site Ozeti)
SVD : Singular Value Decomposition (Tekil Değer Ayrıştırması)
TF : Term Frequency (Terim Frekansı)
URL : Uniform Resource Locator (Tekdüzen Kaynak Bulucu)
XML : Extensible Markup Language (Genişletilebilir Işaretleme Dili)
YSA : Yapay Sinir Ağı

TÜRKÇE HABER METİNLERİNİN YAKLAŞIK EN YAKIN KOMŞU ARAMA YÖNTEMİ İLE BENZERLİK TESPİTİ

ÖZET

İnternet haberciliğinin yaygınlaşması ile birlikte geleneksel medya organları büyük bir dönüşüm geçirmektedir. Çevrimiçi ve çevrimdışı platformlarda yayınlanan haberler çoğunlukla konusu dışında içerikler barındırmaktadır. Aynı zamanda internet üzerinde yayınlanan haberler genellikle benzer içeriklerle farklı kaynaklar tarafından yayınlanmakta ve okuyucuların haberin içeriğine hızlı bir şekilde ulaşması zorlaşmaktadır. Haber sitelerinin daha fazla kullanıcı tarafından ziyaret edilmesi amacıyla tıklama tuzağı (clickbait) başlıklar kullanılmakta ve ziyaretçilerin gazetecilikte 5N1K kuralı olarak adlandırılan; ne, ne zaman, nerede, nasıl ve kim gibi haber öğelerine ulaşması zorlaşmaktadır. Bu çalışmada Türkçe yayın yapan 5 farklı haber kaynağından 350.000 Türkçe haber geliştirilen bir web kazıyıcı (scraper) aracılığıyla toplanmış ve özetlenmiştir. Bu haberler içerisinden 2022 Ocak ayında yayınlanmış toplam 28.000 haber kullanılarak eğitim verisi oluşturulmuştur. Ham haber metinleri ve özet haber metinleri kullanılarak haber vektörleri oluşturulup, haber vektörleri arasında yaklaşık en yakın komşu araması yapılarak farklı kaynaklardan yayınlanan benzer haberlerin tespit edilmesi için bir yöntem geliştirilmiştir. Sistemin başarısının ölçülmesi için web tabanlı bir oylayıcı sistem geliştirilmiş ve uzmanlar tarafından değerlendirilmiştir. Uzman oylaması sonucunda vektörleştirme yöntemlerinin başarısı ve özetlemenin başarı üzerine etkisi karşılaştırılarak performans değerlendirmesi yapılmıştır.

Anahtar Kelimeler: Doc2vec, Metin Benzerliği, Metin Özetleme, Yaklaşık En Yakın Komşu Arama.

SIMILARITY DETECTION OF TURKISH NEWS ARTICLES WITH APPROXIMATE NEAREST NEIGHBOR SEARCH

ABSTRACT

With the spread of internet journalism, traditional media organs are undergoing a huge transformation. The news published on online and offline platforms mostly contain different content other than their subjects. News articles on the internet are generally published by different sources with similar content and it is difficult to reach the content of the news quickly for the visitors. Clickbait headlines are used to be visited by more users. It becomes harder for visitors to reach the basic elements of journalism, such as what, when, where, how and who. In this study, 350.000 news which were published from 5 different news sources in Turkish were collected through a web scraper developed by us and summarized. Training data was created by using 28.000 news published in January 2022 from these news. A method has been developed to detect similar news articles from different sources by creating news vectors using raw and summarized news articles, and approximate nearest neighbors searching among news vectors. To measure the success of the system, a web-based voting system has been developed and evaluated by experts. As a result of expert voting, the success of vectorization methods and the effect of summarization on success were compared and performance was evaluated.

Keywords: Doc2vec, Text Similarity, Text Summarization, Approximate Nearest Neighbor Search.

1. GİRİŞ

İnternet kullanıcı sayısının yüksek bir ivmeyle artması, genişbant internet hizmetlerinin yaygınlaşması ve mobil cihazlarla internete erişen internet kullanıcı sayısındaki artış geleneksel haberciliğin dijital dünyadaki şekline evrilmesi yönünde önemli bir rol oynamıştır. Geleneksel basılı yayınlara olan ilgi günden güne azalmaktadır ve bu gelişmeler sonucunda internet haberciliği kavramı ortaya çıkmıştır. İnternet haberciliği yalnızca internet üzerinden haber yayını yapılması anlamına gelen bir medya terimidir. İnternet haberleri basılı şekilde okuyucuya ulaşmaz ve televizyon haberciliğinde olduğu gibi saati belirli bir haber bülteni yoktur. İnternet haberciliğinin yaygınlaşması ile birlikte medya kuruluşlarında bir dijital dönüşüm gerçekleşmektedir. Medya kuruluşları, medyada meydana gelen bu dönüşüm ile birlikte haber yayınlarının çoğunu internete taşımıştır. İnternet kullanıcıları haber ihtiyaçlarının büyük bir kısmını haber sitelerinden ve sosyal medya uygulamalarından karşılamaktadır. Bununla birlikte bu dönüşüm bazı açılardan medya kuruluşları için dezavantaj oluşturmaktadır.

Haber sitelerindeki gelir modelinin büyük bir kısmı reklamlardan oluşmaktadır. Ancak internet kullanıcılarının haber ihtiyaçlarını karşılamak için haber kaynakları yerine sosyal medya uygulamalarına yönelmesi, reklam engelleme yazılımlarının yaygınlaşması gibi sebeplerle kısır bir döngü oluşmaktadır. Haber sitelerinin ziyaretçi sayısı ve dolayısıyla reklam gelirleri azalırken daha fazla sayıda okuyucuya ulaşması için, internet kullanıcılarının ilgisini çekmek amacıyla haber sitelerinden yayınlanan clickbait haberler yaygınlaşmaktadır. Clickbait; internet kullanıcılarının ilgisini çekecek şekilde oluşturulan başlıklar ve reklam bannerları gibi materyallerle kullanıcının hedef sayfayı görüntülemesini amaçlayan bağlantı türüdür. İnternet haberciliğinde genellikle hedefin ilgisini çekmek amacıyla yarım bırakılmış başlıklar, arama terimlerinden ziyaretçi kazanmak amacıyla soru cümlelerinden oluşan başlıklar veya haber içeriğiyle alakasız başlıklar kullanılarak clickbait yapılmaktadır. İnternet sitelerinin öneminin belirlenmesinde; ziyaretçilerin sayfada geçirdiği süre, toplam sayfa görüntülenme sayısı, ziyaretçinin sayfadan hemen çıkma oranı gibi metrikler önemli olduğu için clickbait bağlantılar üzerinden gelen ziyaretçinin sayfada geçirdiği süre de önem kazanmaktadır. Haber sitelerinden yayınlanan haberlerin büyük bir kısmı haberin içeriğinden farklı başlık ve manşetlerle sunulmaktadır. Farklı kaynaklardan benzer haberler birbirinden

farklı başlık ve manşetlerle sunulmakta ve okuyucuların doğru habere erişmesi zorlaşmaktadır. İnternet kullanıcılarının yanı sıra medya takip ajansları gibi kuruluşlar için de farklı kaynakların taranması zahmetli ve maliyetli bir süreçtir. Bu sebeplerle haber kaynaklarından yayınlanan haberler arasında benzer haberlerin tespit edilmesi önemli bir problemdir.

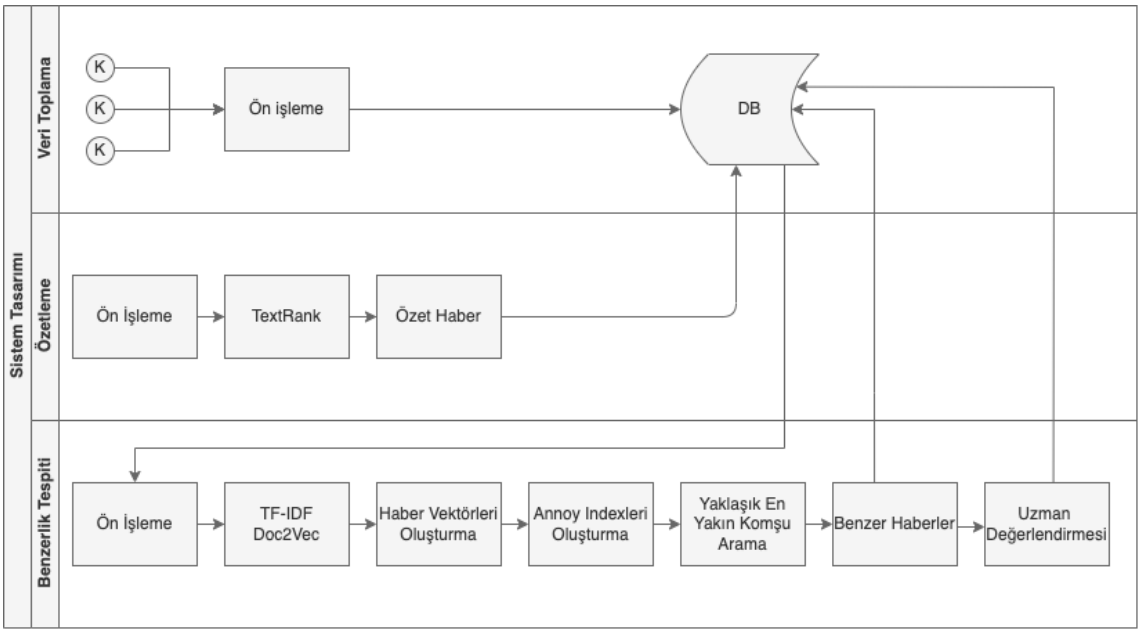
Haber metinlerinin otomatik metin özetleme yöntemleriyle özetlenmesi ile birlikte, hem kullanıcıların ihtiyaç duyduğu haberin konusu ile ilgili kısa ve net bilgilere ulaşması hem de makine öğrenmesi görevlerinde verinin ön hazırlık aşamaları için kolaylık sağlanmaktadır. Otomatik metin özetleme literatürde kendisine geniş bir çalışma alanı bulmaktadır. Metin özetleme ile ilgili ilk çalışmalar erken bilgisayar dönemlerinde başlamış ve henüz mikroçiplerin bilgisayar sistemlerinde kullanılmadığı dönemlerde ilk çalışma Hans Peter Luhn isimli bilim insanı tarafından 1958 yılında yapılmıştır. Luhn metin özetleme sistemlerinde kelime frekanslarının önemini ve özetlenecek metinde yer alan yüksek frekanslı kelimelerin özet metnin oluşturulması sürecinde kullanılabileceğini belirtmiş ve en yüksek frekansa sahip kelimelerin bulunduğu cümlelerin metin için en önemli cümleler olabileceği tezini savunmuştur (Luhn, 1958). Terim frekansları hem özetleme hem de sınıflandırıcılar için önemli bir ölçü birimi olarak çok sayıda algoritma tarafından kullanılmaktadır.

Edmundson ve Wyllys (1961) ise özetleme için metnin başlığında yer alan kelimelerin önemini vurgulamış ve düşük semantik özellikli kelimelerin (ama,yine,zaten vb.) filtrelenmesini önermiştir. Daha sonra Edmundson (1969) tarafından yapılan çalışmada ise cümle seçiminde cümlelerin önemli olabileceği yönünde ipucu olabilecek kelimelerin kullanılması önerilmiştir. Luhn ve Edmundson'ın çalışmaları erken dönemlerde yapılan çalışmalar olmasına rağmen cümlelerin seçilmesi için modern metin özetleme sistemlerinde hala kullanılmaktadır. Mihalcea ve Tarau (2004) tarafından anahtar kelime ve cümle çıkarmak için çizge(*graph*) tabanlı bir sıralama yaklaşımı önerilmiştir. Çizge tabanlı sıralama algoritmaları, Google'ın PageRank algoritması gibi internet sitelerindeki link yapısının analizi alanlarında başarıyla kullanılmaktadır (Brin ve Page, 1998).

Metinlerin özetlenmesi ile birlikte daha küçük boyutlu metinler elde edilebilmekte ancak yüksek sayıda metin verisiyle çalışıldığında, özet metinler kullanılması

durumunda dahi performans problemleri ortaya çıkabilmektedir. Metinler arasındaki benzerliklerin tespit edilmesi için metinlerin vektör uzayında temsil edilmesi gereklidir. Bununla birlikte k-NN gibi en yakın komşu arama algoritmaları çok sayıda ve yüksek boyutlu vektörlerde performans sorunlarıyla karşılaşmaktadır. Bu sorunların çözümüne yönelik olarak yaklaşık en yakın komşu arama algoritmaları öne çıkmaktadır.

Yaklaşık en yakın komşu arama problemi ile ilgili literatürde çok sayıda çalışma bulunmaktadır. Araştırmacılar en yakın komşuların aranması problemi ile ilgili çeşitli algoritmalar geliştirmiş ve yöntemlerden bazıları gerçek hayattaki problemlerin çözümünde kullanılmıştır. Araştırmacılar Spotify'da müzik önerileri için ANNOY isimli yaklaşık en yakın komşu bulma kütüphanesini kullanmıştır (URL-1). Bir başka çalışmada makine öğrenmesi problemlerinde kullanılmak üzere Houle ve Nett (2015) tarafından Rank Cover Tree(RCT) önerilmiş, Sun ve diğ. (2014) tarafından ise veritabanı sorguları için SRS önerilmiştir. En yakın komşu arama problemi için algoritmalar içeren, çok boyutlu uzaylarda en yakın komşu araması için popüler bir yöntem olan LSH (Locality-Sensitive Hashing) tabanlı bir FALCONN kütüphanesi geliştirilmiştir (URL-2). Bir başka çalışmada ise, yaklaşık en yakın komşu araması için C++ programlama dili ile geliştirilen FLANN önerilmiştir (Muja ve Lowe, 2009).



Şekil 1.1. Sistem tasarımı

Bu tez çalışması kapsamında farklı kaynaklardan yayınlanan Türkçe haber metinleri üzerinde, yaklaşık en yakın komşu arama yöntemini kullanarak özet ve ham haber metinlerinde benzerlik tespiti yapılmasını sağlayan bir sistem geliştirilmesi amaçlanmıştır. Çalışma kapsamında geliştirilen sistemlere dair akış Şekil 1.1.'de gösterilmektedir. Sistem veri toplama, özetleme ve benzerlik tespiti olacak şekilde üç ana başlık altında geliştirilmiştir. Geliştirilen sistemler ortak bir veri tabanı üzerinde çalışmaktadır.

Çalışmanın "Makine Öğrenmesi" başlığında makine öğrenmesi kavramı ve makine öğrenmesi yöntemleri ele alınmıştır. "Yaklaşık En Yakın Komşu Arama" başlığında bilgisayar bilimlerinin önemli bir optimizasyon problemi olan en yakın komşu arama problemi, mesafe hesaplamak için kullanılan mesafe ölçütleri ve popüler en yakın komşu arama algoritmaları incelenmiştir. Sonrasında "Metin Özetleme Sistemleri" başlığında, özetleme yaklaşımları ve otomatik metin özetleyicilerin performans değerlendirme metrikleri incelenmiştir. Çalışmanın "Dil modeli geliştirme" başlığında, metinler üzerinde çalışmak için gerekli olan dil modellerinin gelişim süreci ele alınmış, TF-IDF gibi geleneksel yöntemlerden başlanarak sinir ağı modellerini kullanan Doc2Vec gibi modern yöntemler ele alınmıştır.

Çalışmanın "Malzeme ve Yöntem" başlığında ise çalışma kapsamında geliştirilen sistemler hakkında bilgiler verilmiştir. Çalışma kapsamında veri seti olarak kullanılmak üzere, Türkçe yayın yapan haber kaynaklarından 350.000 adet haber, geliştirilen bir web kazıyıcı ile toplanmıştır. Metin özetleyici sistemler için literatürde çokça kullanılan çıkarıma dayalı özetleme yöntemlerinden TextRank algoritması ile haber kaynaklarından toplanan 350.000'den fazla haber otomatik bir şekilde özetlenmiştir ve özetlenen haber metinleri çalışma kapsamında kullanılmak üzere veritabanına kaydedilmiştir. 2022 Ocak ayında yayınlanan toplamda 28.000 haber metninin ham halleri ve özet metinleri kullanılarak TF-IDF ve Doc2Vec yöntemleri ile haber metinlerinin vektör uzayında temsil edilmesine yönelik vektörleştirme işlemleri yapılmıştır. TF-IDF vektörleri ham haber metinleri ve özet haber metinleri olacak şekilde iki ayrı şekilde, Doc2Vec vektörleri ise farklı vektör boyutları ile temsil edilecek şekilde toplamda 6 farklı vektör boyutu ve ham/özet haberler olacak şekilde toplamda 12 farklı şekilde vektörleştirilmiştir. Yaklaşık en yakın komşu arama algoritmalarından

Annoy üzerinde arama yapılması için, vektörleştirilen haber metinleri kullanılarak Annoy vektör indeksleri oluşturulmuş ve dosya sistemi üzerine kaydedilmiştir. Sonrasında ise geliştirilen web tabanlı oylayıcı sistem kullanılarak benzer olarak tespit edilen haberler uzman oylamasına sunulmuş ve uzman değerlendirmelerine dair sonuçlar veritabanına kaydedilmiştir.

Çalışmanın "Bulgular ve Tartışma" başlığında farklı vektörleştirme yöntemleriyle elde edilen haber vektörleri kullanılarak, ayrı ayrı ham haber metinleri ve özet haber metinleri ile yapılan deney sonuçları ve uzman değerlendirmesi sonucunda elde edilen bulgular ele alınmış, "Sonuçlar ve Öneriler" başlığında ise değerlendirmeler ve diğer çalışmacılar için öneriler yapılmıştır.

2. MAKİNE ÖĞRENMESİ

Makine öğrenmesi, veriler ve algoritmalar kullanarak insan öğrenmesini taklit etmeyi hedefleyen, kademeli bir şekilde doğruluğunu artıran bir bilgisayar bilimleri ve yapay zeka çalışma alanıdır.

Makine öğrenmesi terimi ilk olarak 1959 yılında Arthur Samuel tarafından ortaya atılmıştır. Samuel makine öğrenmesini; bilgisayarların açık bir şekilde programlanmadan, onlara öğrenme yeteneği sağlayan bir çalışma alanı olarak tanımlar (Samuel, 1959).

Makine öğrenmesi, gerçek dünyadaki problemlerde kendisine geniş bir kullanım alanı bulmaktadır. Gerçek dünyadaki bazı problemler yüksek karmaşıklığa sahiptir. Klasik programlama yaklaşımlarıyla çözüm sağlanması zor olan bu problemlerde makine öğrenmesi teknikleri ile çözüm sağlanabilmektedir. Makine öğrenmesi; spam e-posta iletilerinin filtrelenmesi, yüz ve nesne tanıma, trafik yoğunluğunun tahmini, dolandırıcılık algılama gibi alanlarda kendisine kullanım alanı bulmaktadır. Diğer yandan; otonom araçlar, sosyal ağlardaki arkadaş önerileri, e-ticaret platformlarındaki ürün önerileri makine öğrenmesinin yine pratikteki uygulama alanlarındandır.

Bir makine öğrenmesi görevinin gerçek hayattaki problemlerin çözümünde kullanılması sürecinde, veri toplama, veri hazırlama, modelin eğitimi, modelin değerlendirilmesi ve modelin geliştirilmesi olmak üzere beş temel adım uygulanmaktadır. Görevin ne olduğundan bağımsız olarak, bir makine öğrenmesi algoritması bu beş adımı takip ederek geliştirilebilir (Lantz, 2019).

Verilerin Toplanması: Veri toplama adımı makine öğrenmesi görevinde kullanılmak üzere ham verileri toplamayı içerir. Metin dosyaları, veritabanları, web sayfaları, haber siteleri, wiki sayfaları, görüntü dosyaları, sosyal ağlar üzerinden elde edilen kullanıcı ilişkileri, kullanıcı yorumları gibi çeşitli kaynaklardan ham veriler toplanır. Çoğu senaryoda çeşitli kaynaklardan toplanmış veriler tek bir kaynak olarak birleştirilir.

Verilerin Hazırlanması: Bir makine öğrenmesi görevinde üretilecek olan sonucun kalitesi, kullanılacak verilerin kalitesine bağlıdır. Bu sebeple, toplanan verilerin bir ön hazırlıktan geçirilmesi önemlidir. Verilerin hazırlanması süreci, karmaşık haldeki

verilerin tasnif edilmesi, gereksiz durak kelimelerin temizlenmesi, kısaltmaların ve noktalama işaretlerinin model için gerekli değilse temizlenmesi gibi birtakım işlemleri kapsar.

Modelin Eğitilmesi: Verilerin toplanması ve hazırlanmasının ardından, bu verilerden nasıl bir bilgi çıkarılabileceği hakkında fikir sahibi olunması muhtemeldir. Modelin eğitilmesi, makine öğrenmesi modelini oluşturmak için en doğru sonucu üretecek olan algoritmanın belirlenmesi işlemini kapsar. Bu aşamada örneğin; gözetimli öğrenme yöntemlerinden birisi kullanılacaksa toplanan veriler eğitim ve test verisi olacak şekilde ayrılabilir. Eğitim verileri kullanılarak probleme uygun bir model oluşturulur.

Modelin Değerlendirilmesi: Makine öğrenmesi modelinin eğitilmesinin ardından, sistemin doğruluğunun test edilmesi gerekmektedir. Test işlemi için, geliştirilen model test verileri üzerinde çalıştırılır. Bu aşamada modelin test verileri üzerindeki başarısını ölçebilmek için elde edilen sonuçlar kaydedilir. Değerlendirme, sonuçların doğruluk, hassaslık ve belirginlik açısından incelenmesiyle elde edilir.

Modelin Geliştirilmesi: Modelin başarısı çok sayıda parametreye bağlı olarak değişkenlik gösterebilmektedir. Geliştirilen modelin başarısının yeterli gelmediği durumlarda farklı stratejiler geliştirilip modelin performansının iyileştirilmesi hedeflenmektedir. Bazı senaryolarda modelin tamamen değiştirilmesi, daha fazla veya az sayıda değişkenle verinin tekrardan hazırlanması gerekmektedir.

2.1. Makine Öğrenmesi Yöntemleri

2.1.1. Gözetimli Öğrenme

Gözetimli öğrenme giriş-çıkış çiftlerine dayalı olarak bir girişin bir çıkışa eşleştirildiği makine öğrenmesi görevidir (Russell ve Norvig, 2010). Gözetimli öğrenmede, eğitim verisi olarak sistemin öğrenmesi istenen kavramlar ile ilgili toplanan veriler, bir ön işleme sürecinden sonra etiketlenerek sisteme verilir. Eğitim verisindeki her bir örnek veri, çıktı değerleriyle eşleştirilir. Bu bilgilerle oluşturulan giriş-çıkış ilişkisi ile gelecekte karşılaşılacak gözlemlerin çıktıları tahmin edilebilir. Gözetimli öğrenme bir dizi eğitim örneğinden oluşan etiketlenmiş eğitim verilerinden bir sonuç üretir (Mohri ve diğ., 2012). Gözetimli öğrenme problemleri genel olarak, sınıflandırma ve regresyon

problemleri olacak şekilde iki tür olarak ele alınmaktadır. Metin yazarının cinsiyet tahmini, metin kategorisinin tahmin edilmesi gibi problemler denetimli öğrenmenin bir problemi olarak ele alınmaktadır. Gözetimli öğrenme algoritmaları; destek vektör makineleri, Lineer regresyon, Lojistik regresyon, Naive Bayes, Karar ağaçları, K-en yakın komşu ve yapay sinir ağları olarak sınıflandırılmaktadır.

2.1.2. Gözetimsiz Öğrenme

Gözetimsiz öğrenme yalnızca giriş değişkenlerinden oluşan ve giriş-çıkış ilişkisi bulunmayan bir modele dayanan makine öğrenmesi görevidir. Gözetimsiz öğrenme etiketsiz eğitim verilerini kullanır. Gözetimsiz öğrenmede kullanılan algoritmalarla, sisteme girdi olarak verilen veriler arasındaki ilişkilerin bulunması hedeflenmektedir. Bu öğrenme modelinde, eğitim verilerinden veriler arasındaki ilişkileri ifade edecek bir model geliştirilir. Geliştirilen model sisteme sonradan gelecek giriş değerlerini kullanarak bir sonuca ulaşmaya çalışır. Gözetimsiz öğrenmede çıkış değerleri olmadığı için giriş ve çıkış değerleri arasında bir bağıntı kurulamaz. Bu yüzden sistemin çıkış hakkında bir bilgisi yoktur. Gözetimsiz öğrenme problemleri; birleştirme ve kümeleme problemleri olarak ele alınmaktadır. Gözetimsiz öğrenme, eğitim verisinde bir etiketleme olmaksızın çalışır. Örnek olarak, sosyal ağlarda kullanıcıların ilgi alanlarına göre kümelenmesi ve kullanıcılara tanıyor olabileceği kişilerin önerildiği sistemler verilebilir. Bu örneklerde sistem bir sebep sonuç ilişkisi aramaksızın, yalnızca eğitim verisinden yararlanarak kümeleme işlemi yapar.

2.1.3. Yarı Gözetimli Öğrenme

Yarı gözetimli öğrenme sınıflandırıcıların özel bir formudur. Geleneksel sınıflandırıcılar modelin eğitilmesi için etiketli verilere ihtiyaç duyarlar. Bununla birlikte, etiketli örnekler, deneyimli insan yorumcuların çabalarını gerektirdiğinden, elde edilmesi genellikle zor, pahalı veya zaman alıcıdır. Bu arada, etiketlenmemiş verilerin toplanması nispeten kolay olabilir, ancak bunları kullanmanın birkaç yolu vardır. Yarı denetimli öğrenme, daha iyi sınıflandırıcılar oluşturmak için etiketlenmiş verilerle birlikte büyük miktarda etiketlenmemiş veri kullanarak bu sorunu çözer. Yarı denetimli öğrenme daha az insan çabası gerektirdiğinden ve daha yüksek doğruluk sağladığından hem teoride hem de pratikte büyük ilgi görmektedir (Zhu, 2005).

2.1.4. Pekiştirmeli Öğrenme

Pekiştirmeli öğrenme, denetimli öğrenmeye benzer bir makine öğrenmesi modelidir. Pekiştirmeli öğrenmede algoritma bir örnek veri seti kullanılarak eğitilmez. Pekiştirmeli öğrenmede model, deneme yanılma yoluyla öğrenmeye çalışır.

Pekiştirmeli öğrenmenin kökeni, sibernetikle ilgili ilk çalışmalara kadar uzanmaktadır. Pekiştirmeli öğrenme, istatistik, psikoloji, sinirbilim ve bilgisayar bilimleri içerisinde kendisine uygulama alanı bulmaktadır. Pekiştirmeli öğrenmenin amacı, bir görevin nasıl başarılacağını belirtmeye gerek kalmadan ajanları ödül ve ceza ile programlamanın bir yolunun bulunmasıdır. Ancak bu amacın sağlanması için zorlu hesaplama engelleri bulunmaktadır (Kaelbling ve diğ., 1996). IBM tarafından geliştirilen Watson isimli yapay zeka programı, 2011 yılında Jeopardy! isimli televizyon programındaki meydan okumasında pekiştirmeli öğrenmeyi kullanarak başarıya ulaşmıştır.

2.2. Yapay Sinir Ağları

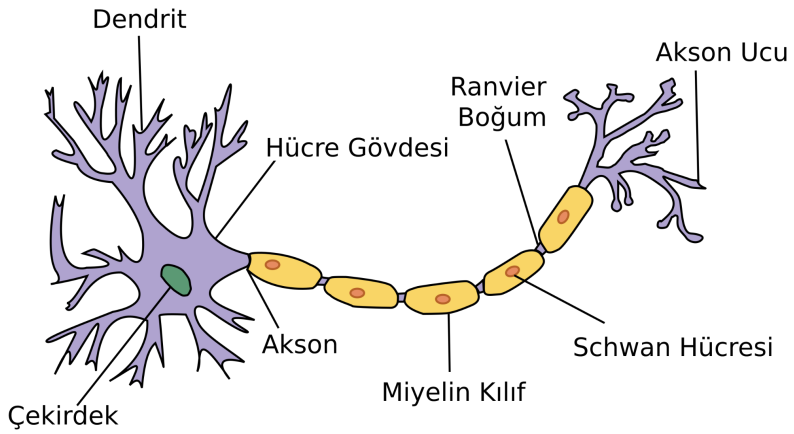
Yapay sinir ağları, insan beyninin öğrenmesini taklit ederek kendi kendine öğrenen sistemler geliştirilmesini hedefleyen bir bilgisayar bilimleri ve yapay zeka çalışma alanıdır. İnsan beyninin öğrenmesi, birbirleriyle karmaşık bir şekilde sinaptik bağlar kuran nöronlar vasıtasıyla gerçekleşir. Yapay sinir ağları ile insan beyninde yer alan biyolojik sinir sistemi, dijital olarak modellenmeye çalışılmaktadır.

Yapay sinir ağlarının ortaya çıkışı, insan beyninin öğrenme sürecinin matematiksel olarak modellenmesi ile başlamıştır. Biyolojik sinir sistemlerinde öğrenme, nöronlar arasında var olan sinaptik bağlantılar vasıtasıyla gerçekleşir. Bir biyolojik sinir sisteminde, sinaptik bağlantılar arasındaki sinyallerin sürekli olarak güncellenmesi canlının öğrenmesini ifade etmektedir.

Yapay sinir ağlarında genellikle biyolojik sinir sistemleriyle olan paralelliklerine atıfta bulunulur. Ancak biyolojik sinir sistemleri ile ilgili hala çok az şey bilinmektedir. İnsan beyninin ve sinir sisteminin modellenmesiyle ilgili çalışmalara olan ilgi her geçen gün artmaktadır. Bununla birlikte yapılan çalışmalardan elde edilen bulgular göz önüne alındığında yapay sinir ağları biyolojik sinir sistemlerinin aşırı derecede basitleştirilmiş bir modeli olarak karşımıza çıkmaktadır. Biyolojik bir sinir sistemine sahip olan insan

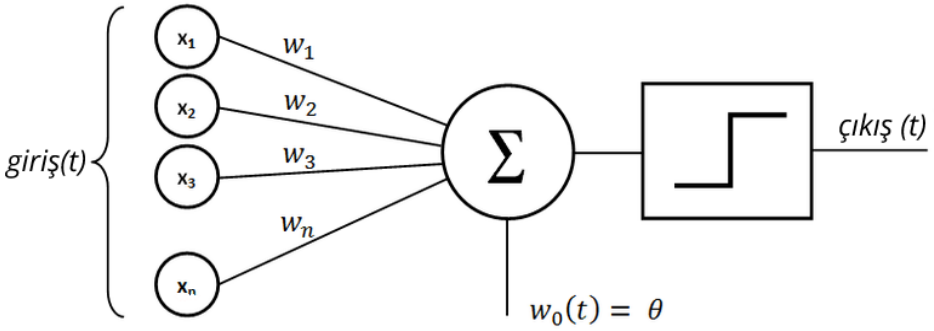
vücudu çok karmaşık sinir ağları ile yaşamını ve öğrenme sürecini sürdürmektedir. Diğer tüm canlılara kıyasla insan, doğumundan itibaren uzunca bir süre bakıma ve öğrenmeye muhtaçtır. İnsan beyni, duyu organları ve biyolojik sensörleri vasıtasıyla dış dünyadan elde ettiği girdileri nöronlar vasıtasıyla bir dizi işlemden geçirerek çıktılar üreterek öğrenir.

Biyolojik bir sinir sisteminde girdiler duyu organları vasıtasıyla elde edilen sinyallerdir. Girdi sinyalleri kullanılarak, nöronlar arasındaki sinaptik bağlantılar sürekli olarak güncellenir ve çıktılar üretilir. Bir yapay sinir ağında ise öğrenme, girdiler ve çıktılar kullanılarak sistemin eğitilmesi ile gerçekleşir. Biyolojik sinir hücresinin temsili gösterimi Şekil 2.1.'de gösterilmektedir.



Şekil 2.1. Biyolojik sinir hücresinin temsili gösterimi (URL-3)

Yapay sinir ağlarında girdiler ve çıktılar arasındaki ilişki matematiksel olarak modellenir. Eğitim verileri giriş ve çıkış değerleri olarak ele alınır ve giriş değerlerinin ağırlıkları ayarlanarak çıkışa en yakın değere yakınsanana kadar eğitim devam eder.



Şekil 2.2. Yapay sinir ağı gösterimi

Bir yapay sinir ağını oluşturan nöronlar, biyolojik sinir sistemindeki sinapslar gibi birbirlerine sinyal iletebilir. Bir nöron kendisine gelen sinyalleri alır, bu sinyalleri işleyerek kendisine bağlı olan diğer nöronlara iletir. Şekil 2.2.'de bir yapay sinir ağı gösterilmektedir. Bir yapay sinir ağındaki girdileri ifade eden sinyaller gerçek birer sayıdır. Her bir nöron kendisine gelen sinyallerin toplamını doğrusal olmayan bir fonksiyon ile hesaplar. Toplama işleminin yapıldığı fonksiyonlar toplayıcı fonksiyon olarak isimlendirilmektedir. Nöronlar arasındaki bağlantılara kenar denir. Yapay sinir ağına gelen sinyaller ağırlıklara sahiptir ve bu ağırlıklar sistemin öğrenme sürecinde ayarlanır.

Toplayıcı fonksiyonlar belli ağırlık ve bias değerleri ile bir yapay sinir ağındaki giriş değerlerinin hesaplandığı fonksiyonlardır. Toplayıcı fonksiyonlar; giriş değerlerinin toplanması, giriş değerlerinin çarpılması, giriş değerleri içerisinden maksimum değerin belirlenmesi, giriş değerleri içerisinden minimum değerin belirlenmesi, giriş değerlerinin kümülatif toplamının alınması gibi işlemleri gerçekleştirebilir. Toplayıcı fonksiyonlar içerisinden en yaygın olarak kullanılan fonksiyon ağırlıklı toplamadır. Bazı toplayıcı fonksiyonlar ve fonksiyonların açıklamaları Tablo 2.1.'de gösterilmektedir.

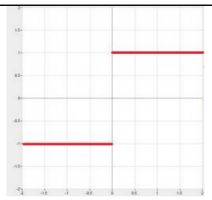
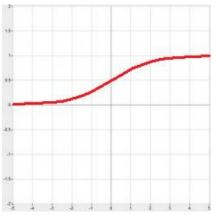
Tablo 2.1. Bazı toplayıcı fonksiyonlar

Adı	**Fonksiyon**	**Açıklama**
Toplam	$Net = \sum_{i=1}^{N} X_i * W_i$	Ağırlık değerlerinin girdilerle çarpılması sonucunda elde edilen değerler birbirleriyle toplanarak Net girdi hesaplanır.
Çarpım	$Net = \prod_{i=1}^{N} X_i * W_i$	Ağırlık değerlerinin girdilerle çarpılması sonucunda elde edilen değerler birbirleriyle çarpılarak Net girdi hesaplanır.
Maksimum	$Net = Max(X_i * W_i)$	n adet girdinin ağırlıkları ile girdilerin çarpılması sonucunda elde edilen değerlerden en büyüğü Net girdi olarak kabul edilir.
Minimum	$Net = Min(X_i * W_i$	n adet girdinin ağırlıkları ile girdilerin çarpılması sonucunda elde edilen değerlerden en küçüğü Net girdi olarak kabul edilir.

Yapay sinir ağına gelen giriş değerlerinin toplayıcı fonksiyon ile hesaplanmasının ardından elde edilen çıkış değeri aktivasyon fonksiyonuna giriş değeri olarak verilir.

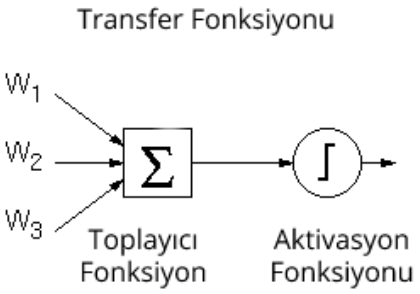
Aktivasyon fonksiyonu hesaplanan toplam değeri kullanarak, bilginin bir sonraki nörona iletilip iletilmeyeceğine karar veren fonksiyondur.

Yapay sinir ağlarında yaygın olarak step fonksiyonu, sigmoid fonksiyonu, tanh fonksiyonu gibi fonksiyonlar aktivasyon fonksiyonu olarak kullanılmaktadır. Step fonksiyonu ve sigmoid fonksiyonunun gösterimi ve açıklamaları Şekil 2.3.'de gösterilmektedir.

Adım (Step) Aktivasyon Fonksiyonu		$F(Net)=\begin{cases}1 & \text{if Net>Eşik Değer}\\ 0 & \text{if Net<=Eşik Değer}\end{cases}$	Gelen Net girdinin belirlenen bir eşik değerin altında veya üstünde olmasına göre hücrenin çıktısı 1 veya 0 değerini alır.
Sigmoid Aktivasyon Fonksiyonu		$F(Net)=\frac{1}{1+e^{-Net}}$	Sigmoid aktivasyon fonksiyonu sürekli ve türevi alınabilir bir fonksiyondur. Doğrusal olmayışı dolayısıyla yapay sinir ağı uygulamalarında en sık kullanılan fonksiyondur. Bu fonksiyon girdi değerlerinin her biri için 0 ile 1 arasında bir değer üretir.

Şekil 2.3. Bazı aktivasyon fonksiyonları (Çayıroğlu, 2015)

Yapay sinir ağlarında toplayıcı fonksiyonla aktivasyon fonksiyonunun bir araya gelerek oluşturduğu yapıya transfer fonksiyonu adı verilmektedir. Şekil 2.4.'te bir transfer fonksiyonu gösterilmektedir.

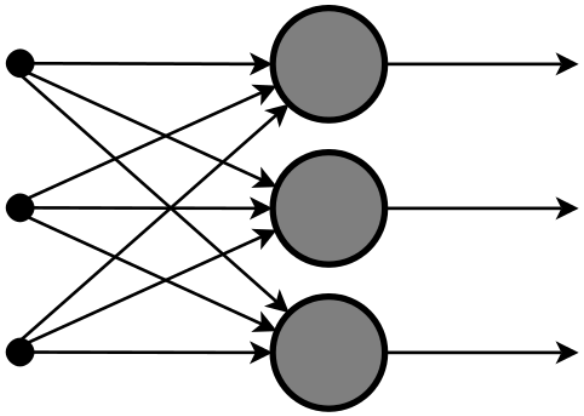


Şekil 2.4. Transfer fonksiyonu

Yapay sinir ağları problemin türüne göre bir veya birden fazla katmana sahip olacak şekilde tasarlanmaktadır. Yalnızca giriş ve çıkış katmanlarına sahip yapay sinir ağları tek katmanlı yapay sinir ağı, giriş ve çıkış katmanları haricinde gizli katman içeren yapay sinir ağlarına ise çok katmanlı yapay sinir ağı ismi verilmektedir. Yapay sinir

ağlarında ele alınacak olan problem doğrusal olarak ayrılabilmesine göre değerlendirilmektedir.

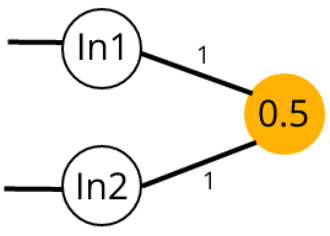
Tek katmanlı yapay sinir ağı, yalnızca giriş ve çıkış katmanlarından oluşan bir sinir ağı modelidir. Tek katmanlı yapay sinir ağlarında gizli katman bulunmaz. Bağlantıların birer ağırlığı vardır ve çıkış katmanı doğrudan giriş katmanındaki tüm girdi değerlerine bağlıdır. Tek katmanlı yapay sinir ağlarında, tek katmanlı ileri beslemeli bir yapay sinir ağı olarak perceptron kavramı önerilmiştir. Şekil 2.5.'de tek katmanlı yapay sinir ağı modeli gösterilmektedir.



Şekil 2.5. Tek katmanlı yapay sinir ağı modeli (URL-4)

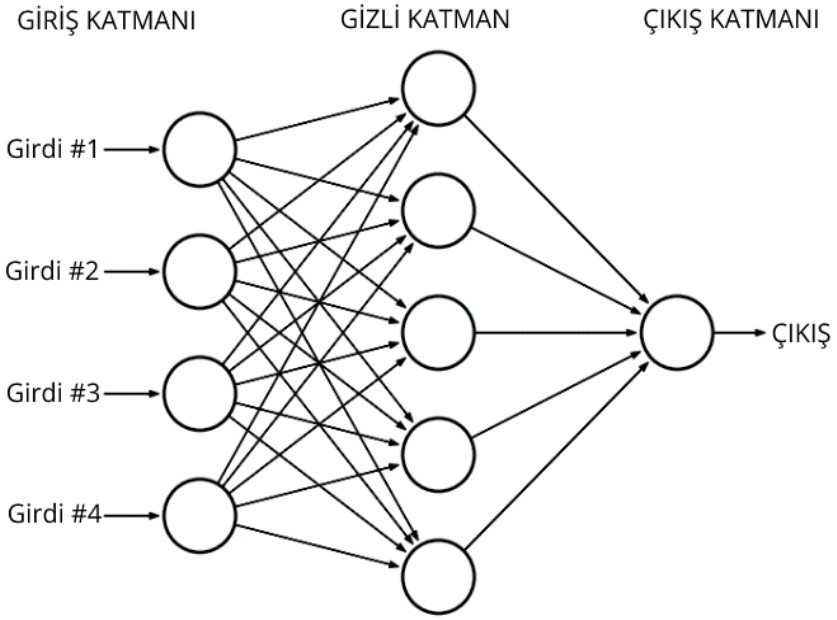
Perceptron 1958 yılında önerilmiş ve yapay sinir ağlarının temel olarak modellenmesi sağlanmıştır. Tek katmanlı ve ileri beslemeli bir yapay sinir ağı olan perceptron, lineer olarak ayrılabilen fonksiyonları öğrenebilir. Bir yapay sinir ağı AND, OR, NOT gibi mantıksal işlemleri perceptron ile öğrenebilir. N adet girdiye sahip bir perceptron çıktı olarak bir değer elde eder. Şekil 2.6.'da OR kapısının yapay sinir ağı ile modellenmesi gösterilmektedir.

In1	In2	Output
0	0	0
1	0	1
0	1	1
1	1	1



Şekil 2.6. OR kapısının YSA ile modellenmesi

Doğrusal olarak ayrılamayan problemlerde tek katmanlı yapay sinir ağları öğrenme işlemini gerçekleştirememektedir. Bu tarz problemlerde yapay sinir ağında gizli katmana ihtiyaç duyulmaktadır. Şekil 2.7.'de giriş katmanı, gizli katman ve çıkış katmanından oluşan çok katmanlı yapay sinir ağı modeli gösterilmektedir.



Şekil 2.7. Çok katmanlı yapay sinir ağı modeli

Çok katmanlı yapay sinir ağları yapay sinir hücrelerinin bir araya gelmesiyle oluşur. Bir çok katmanlı yapay sinir ağı, giriş katmanı, gizli katman ve çıkış katmanı olacak şekilde üç katmandan oluşur. Giriş katmanı, bir yapay sinir ağına gelen girdilerin alındığı katmandır. Giriş katmanı üzerinde herhangi bir değişiklik yapılmadan gizli katmana iletilir. Gizli katman, giriş katmanından iletilen verilerin transfer edildiği katmandır. Bir yapay sinir ağında öğrenme işlemi gizli katman üzerinde gerçekleştirilir. Çıkış katmanı ise, yapay sinir ağına verilen giriş değerlerinden öğrenmesi beklenen çıkış değerlerinin bulunduğu katmandır. Çok katmanlı ağlarda öğrenme en küçük kareler yöntemine dayanan genelleştirilmiş delta kuralını kullanarak gerçekleştirilmektedir.

Yapay sinir ağlarında, giriş katmanından alınan girdilerin önce gizli katmana daha sonra ise çıkış katmanına iletilerek oluşturulan sinir ağı modeline ileri beslemeli ağ denilmektedir. İleri beslemeli yapay sinir ağlarında bilgi yalnızca ileri yönde hareket eder. Bu ağlarda bağlantıların ağırlıkları ağın eğitilmesi sürecinde güncellenebilir ancak geriye doğru bir bağlantı ağırlıklarının güncellenmesi söz konusu değildir.

Geri beslemeli yapay sinir ağları, çıkış katmanları ve ara katmanlardaki çıktı değerlerinin önceki ara katmanlara ve giriş katmanına iletilmesiyle oluşturulan sinir ağı modelidir. Geri beslemeli ağlarda, çıktı değerleri ile gerçek sistem çıktısı arasındaki hata oranları karşılaştırılır ve geri besleme yapılır.

2.3. K-en Yakın Komşu Algoritması

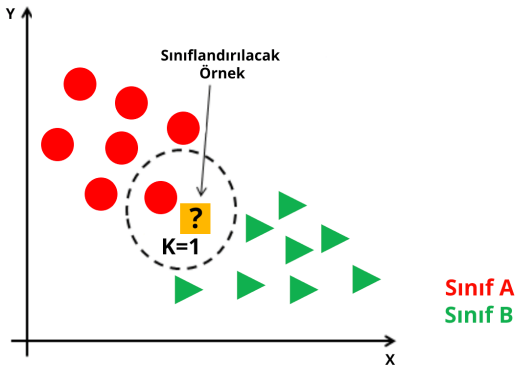
K-en yakın komşu (K-nearest neighbors/k-NN) algoritması gözetimli bir makine öğrenmesi algoritmasıdır. 1967 yılında Cover ve Hart tarafından önerilen algoritma sınıflara ayrılmış verilere sahip veri seti üzerinde, yeni eklenen verinin hangi sınıfa ait olduğunu bulmayı amaçlar (Cover ve Hart, 1967).

K-en yakın komşu algoritması örnek tabanlı bir sınıflandırma algoritmasıdır. K-en yakın komşu algoritması ile sınıflandırmada, öğrenme işlemi eğitim veri setinde bulunan verilere dayanarak gerçekleştirilmektedir. Sınıflandırılacak olan veri, eğitim veri setindeki verilere olan uzaklığına göre sınıflandırılmaktadır.

K-en yakın komşu algoritmasında eğitim verileri, sayısallaştırılarak n boyutlu bir uzayda gösterilir. Her bir eğitim verisi n boyutlu uzayda bir noktayı ifade edecek şekilde belirlenir. Sınıflandırılacak olan verinin örnek uzayda yer alan noktalardan k tanesine olan uzaklığına bakılır. Sınıflandırılacak veri, k tane örnekten en fazla sayıda yakın olduğu etiketle sınıflandırılır. K-en yakın komşu algoritmasında sınıflandırılacak olan verinin öncelikle en yakın k komşusunun belirlenmesi gerekmektedir. Örnek verinin, sınıf verileriyle arasındaki uzaklıkların eşit çıkması durumunda sınıf etiketi belirlenemeyeceği için k değeri genellikle çift bir sayı olarak seçilmez, 1'den büyük tek bir tam sayı olarak seçilir.

k değeri belirlendikten sonra, sınıflandırılacak olan verinin en yakınındaki k komşusuna olan uzaklıkları tek tek hesaplanır. Uzaklık olarak en yakın mesafede bulunduğu komşuların etiketi, sınıflandırılacak verinin etiketi olarak belirlenir ve sınıflandırma tamamlanır. Bu aşamadaki belirleyici unsur k sayıdaki en yakın komşunun sınıf etiketlerinin çoğunluğudur. Şekil 2.8.'de k-en yakın komşu algoritması kullanılarak sınıf etiketinin belirlenmesi işleminde örnek bir eğitim verisinin görselleştirmesi gösterilmektedir. Şekilde kırmızı daireler ve yeşil üçgenlerin ifade ettiği iki farklı

sınıftan örnekler bulunmaktadır. Yeni bir örnekle karşılaşıldığında, iki sınıfa ait verilerin bulunduğu eğitim verileri kullanılarak yeni örneğin hangi sınıfa ait olduğu belirlenmeye çalışılmaktadır.

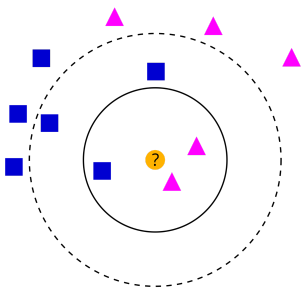


Şekil 2.8. K-en yakın komşu algoritması eğitim verisi örnek gösterim

3. YAKLAŞIK EN YAKIN KOMŞU ARAMA

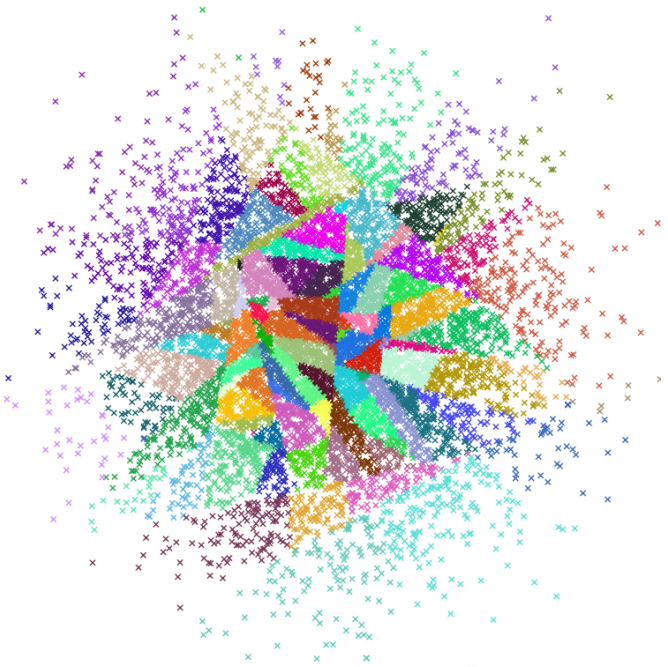
En yakın komşu arama (Nearest Neighbor Search / NNS) problemi bilgisayar bilimlerinde, belirli bir küme içerisinde belirli bir noktaya en yakın veya en benzer noktaları bulmanın amaçlandığı bir optimizasyon problemidir. En yakın komşu araması, bilgisayar bilimlerinde olduğu gibi mühendisliğin birçok alanında kendisine uygulama alanı bulan önemli bir problemdir. En yakın komşu arama problemi bilgisayar bilimlerinde; görüntü tanıma, veri sıkıştırma, örüntü tanıma ve sınıflandırma, istatistiksel sınıflandırma, bilgisayarla görü gibi çeşitli uygulamalar için büyük önem taşımaktadır. Ancak bu problemin yüksek boyutlu uzaylarda çözmek zor bir iştir ve standart kaba kuvvet arama algoritmalarından önemli ölçüde daha iyi bir performans gösteren bir algoritma bulunmamaktadır. Bu durum, yaklaşık en yakın araması gerçekleştiren algoritmalara olan ilginin artmasına yol açmıştır.

En yakın komşu arama problemi Knuth tarafından posta ofisi problemi (post-office problem) olarak tanımlanmıştır. En yakın komşu arama problemi; n nokta kümesi ve d boyutlu bir uzayda, yeni bir Q sorgu noktası verildiğinde, Q sorgu noktasının n nokta kümesi içerisindeki noktalardan Q'ya eşit uzaklıkta veya daha yakın uzaklıkta olan noktaların bulunmasıdır (Knuth, 1973). En yakın komşu aramanın genelleştirilmiş formu, k adet en yakın komşu noktayı bulmayı sağlayan K-en yakın komşu algoritmasıdır. K-en yakın komşu algoritması ile bir veri kümesindeki bir sorgu vektörüne en çok benzeyen vektörler bulunabilir. Veri kümesi içindeki vektör temsilleri ile eşleştirilen K-en yakın komşu arama sorguları ile; bir metin içerisinde bir sorunun cevabını içermesi muhtemel paragrafların bulunması, büyük bir veri kümesi içerisinde tekrarlayan görüntülerin algılanması, belirli bir şarkıya benzeyen şarkıların bulunması gibi görevler gerçekleştirilebilmektedir.



Şekil 3.1. K-en yakın komşu algoritmasının tipik bir görselleştirmesi

K-en yakın komşu algoritmasında düşük boyutlu vektörler arasında başarıyla çalışan, KD-ağaçları gibi veri yapıları bulunmaktadır. Şekil 3.1.'de K-en yakın komşu algoritmasının tipik bir görselleştirilmesi gösterilmektedir. K-en yakın komşu algoritması ile düşük boyutlu vektörler arasında arama sorguları başarıyla çalışmaktadır. Ancak modern kelime ve paragraf gömme modellerinde metin ve görsellerin vektörleştirilmesi sonucunda genellikle yüksek boyutlu vektörler elde edilmektedir. Bunun sonucunda, yüksek boyutlu vektörlerden oluşan vektör uzaylarında hızlı bir şekilde en yakın komşuların aranması ciddi bir problem olarak karşımıza çıkmaktadır. K-en yakın komşu algoritması gibi en yakın komşu arama algoritmalarında en doğru sonucun elde edilmesinin hedeflenmesi gibi sebeplerle performans kayıpları yaşanmakta ve arama sorgularında hız problemleriyle karşılaşılmaktadır. Bununla birlikte yaklaşık en yakın komşu arama algoritmaları ile hız sorununun üstesinden gelinmektedir. Ancak yaklaşık en yakın komşu arama algoritmaları ile tespit edilen k adet en yakın komşu vektörleri her zaman doğru vektörler olamayabilmektedir. Yine de, yaklaşık en yakın komşu arama algoritmaları yüksek boyutlu vektörler arasında efektif bir şekilde çalışmakta ve yüksek boyutlu veri setleri için iyi bir performans göstermektedir. Şekil 3.2.'de yaklaşık en yakın komşu aramasının vektör uzayda örnek bir görselleştirmesi gösterilmektedir.



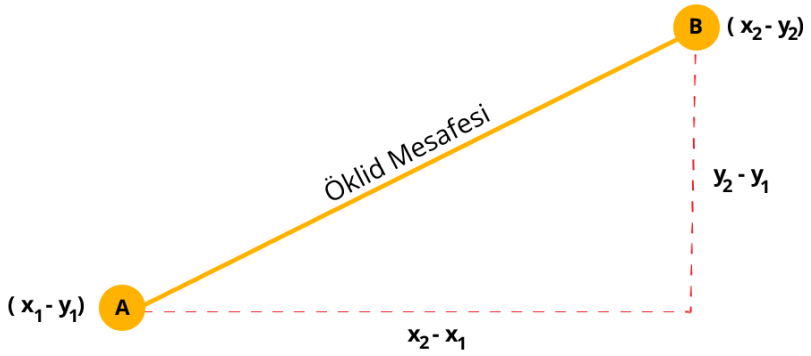
Şekil 3.2. Yaklaşık en yakın komşu aramasının vektör uzayında görselleştirilmesi (URL-1)

3.1. Mesafe Ölçütleri

En yakın komşu arama problemlerinde, iki nokta arasındaki uzaklığı ölçmek için mesafe hesaplama yöntemleri kullanılmaktadır. En yakın komşu arama problemlerinde, problemin türüne göre farklı mesafe hesaplama yöntemlerinden probleme uygun olanı belirlenerek kullanılmaktadır. Metin verileriyle çalışılması durumunda, metin benzerliklerinin tespit edilmesi için literatürde yaygın olarak kosinüs benzerliği kullanılmaktadır.

3.1.1. Öklid Mesafesi

Öklid mesafesi, sınıflandırma algoritmalarında yaygın olarak kullanılan bir mesafe ölçütüdür.



Şeki 3.3. Öklid mesafesinin gösterimi

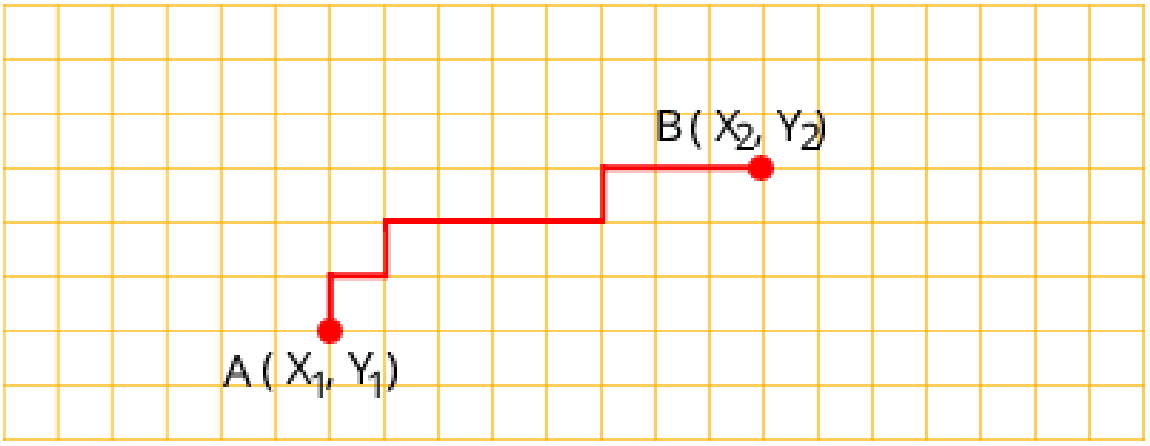
Şekil 3.3.'te A ve B iki farklı nokta olacak şekilde, A ve B noktaları arasındaki Öklid mesafesinin hesaplanması gösterilmektedir.

Öklid mesafesi ile iki nokta arasındaki doğrusal uzaklık hesaplanmaktadır. N boyutlu bir uzayda A ve B noktaları arasındaki Öklid mesafesi, $A=\{x_1,y_1\}$ ve $B=\{x_2,y_2\}$ olmak üzere denklem (3.1)'e göre hesaplanır.

$$\left(\sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}\right) \tag{3.1}$$

3.1.2. Manhattan Mesafesi

Manhattan mesafesi ile iki nokta arasındaki uzaklık belirlenirken, noktalar arasındaki farkların mutlak değerlerinin toplamı hesaplanmaktadır. Şekil 3.4.'te A ve B noktaları arasındaki Manhattan mesafesinin hesaplanması gösterilmektedir.



Şekil 3.4. Manhattan mesafesinin gösterimi

Manhattan mesafesi, $A=\{x_1,x_2,x_3,x_4\ldots,x_n\}$ ve $B=\{y_1,y_2,y_3,y_4\ldots,y_n\}$ olmak üzere denklem (3.2)'ye göre hesaplanır.

$$\left(\sum_{i=1}^{n}\left|x_i - y_i\right|\right) \tag{3.2}$$

3.1.3. Minkowski Mesafesi

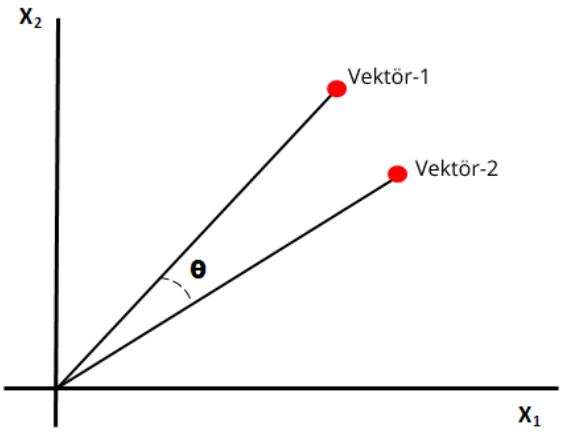
Minkowski mesafesi, sınıflandırma problemlerinde yaygın bir şekilde kullanılan Manhattan mesafesi, Öklid mesafesi gibi ölçütlerin genel bir formudur. Minkowski mesafesi, A={x1,x2,x3,x4…,xn} ve B={y1,y2,y3,y4…,yn} olmak üzere denklem (3.3)'e göre hesaplanır.

$$\left(\sum_{i=1}^{n}\left|x_i - y_i\right|^{p}\right)^{1/p} \tag{3.3}$$

Minkowski mesafesi genel bir formülle ifade edilmektedir ve p değerine göre bazı özel durumları da formülize etmektedir. Minkowski mesafesinde, p=2 olduğu durumda Öklid mesafesi, p=1 olduğunda ise Manhattan mesafesi hesaplanmaktadır.

3.1.4. Kosinüs Benzerliği

Kosinüs benzerliği metin madenciliği ve sınıflandırma problemlerinde yaygın bir şekilde kullanılan bir benzerlik ölçütüdür. Kosinüs benzerliği ile iki vektör arasındaki benzerlik hesaplanmaktadır. Metinlerin benzerliği için ele alınacak olursa, sayısallaştırılmış ve bu sayede vektör olarak ifade edilmiş iki metin arasındaki benzerliği hesaplamak için kullanılmaktadır.



Şekil 3.5. Kosinüs benzerliğinin gösterimi

Kosinüs benzerliği 0 ile 1 arasındaki sınırlı bir aralıkla ifade edilen bir değerdir. Benzerlik için iki vektör arasındaki açının kosinüs değerini hesaplayarak çalışır. İki vektör arasındaki açının 90 derece olduğu durumda bu iki vektör birbirlerine dik vektörler olacaktır ve bu durumda kosinüs benzerliği 0 olacaktır. İki vektör arasındaki açı 0 derece olduğunda ise bu iki vektör aynı yönlü vektörler olacak ve kosinüs benzerliği 1 olacaktır. Yani kosinüs benzerliği 1 değerine doğru yaklaştıkça iki vektör arasındaki açı küçülür.

Şekil 3.5.'te iki vektör arasındaki kosinüs benzerliği gösterilmektedir. Şekildeki vektör-1 ve vektör-2 vektörlerinin kosinüs benzerliği, aralarındaki açının kosinüsüdür. İki vektör arasındaki kosinüs benzerliği, A ve B iki ayrı vektör olacak şekilde denklem (3.4)'te gösterildiği şekilde hesaplanmaktadır.

$$similarity = cos(\theta) = \frac{A \cdot B}{||A||\,||B||} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \quad (3.4)$$

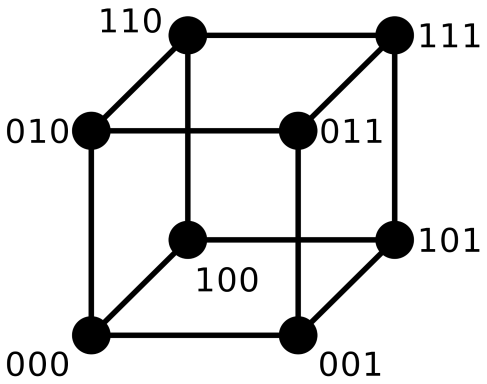
3.1.5. Hamming Mesafesi

Hamming mesafesi bilgisayar bilimlerinde, iki eşit uzunluktaki karakter dizisi arasında aynı pozisyonda bulunan karakterlerden kaç tanesinin farklı olduğunun ölçüsüdür. Bir diğer deyişle eşit uzunluktaki iki karakter dizisinden birinin diğerine dönüşmesi için gerekli olan yer değiştirme sayısıdır. Hamming mesafesi hata algılama ve hata düzeltme için yaygın bir şekilde kullanılmakla birlikte, metin madenciliği ve makine öğrenmesi problemlerinde de sıklıkla kullanılmaktadır. Tablo 3.1.'de farklı karakter dizileri arasındaki hamming mesafelerinin hesaplanmasına ilişkin örnekler gösterilmektedir.

Tablo 3.1. İki karakter dizisi arasındaki Hamming mesafeleri

Karakter Dizisi-1	**Karakter Dizisi-2**	**Hamming Mesafesi**
yavaş	**s**avaş	1
mal**tepe**	mal**zeme**	4
ak**asya**	ak**baba**	4
11001**00**	11001**11**	2

Hamming mesafesi ile binary(ikilik) vektörler arasındaki mesafe hesaplaması da yapılabilmektedir. Aynı uzunluğa sahip iki binary vektör arasındaki hamming mesafesi ile iki vektör arasındaki farklı olan bitlerin sayısı hesaplanabilir.



Şekil 3.6. Küp üzerinde Hamming mesafesi

Şekil 3.6.'da Hamming mesafesinin hesaplanması için 3 bitlik binary bir küp gösterilmektedir. Küpün her bir köşesi 3 bitlik bir karakter dizisini ifade etmektedir. İki köşe arasındaki minimum mesafe ise köşelerde ifade edilen binary karakter dizileri arasındaki Hamming mesafesidir.

3.2. Yaklaşık En Yakın Komşu Arama Kütüphaneleri

3.2.1. ANNOY

Annoy (Approximate Nearest Neighbors Oh Yeah) çok boyutlu bir uzayda, bir sorgu noktasına yakın noktaları aramak için geliştirilmiş bir C++ kütüphanesidir. Literatürde yaklaşık en yakın komşuların bulunması için farklı algoritma ve kütüphaneler bulunmaktadır. Annoy düşük bellek tüketimi, işlemler arasında belleğin paylaşılabilmesi, yüksek boyutlu verilerde gösterdiği performans ile yaygın olarak kullanılan kütüphaneler arasında öne çıkmaktadır (URL-1).

Annoy kütüphanesinin önemli bir avantajı indeksleri statik dosyalar halinde dosya sistemi üzerinde saklayabilmesidir. Bu sayede indeksler çok sayıda farklı işlem tarafından kullanılabilmektedir. Annoy kütüphanesi bunun için dizinlerin bellek üzerinde adreslenmesini sağlayan mmap isimli Unix sistem çağrısını kullanmaktadır.

En yakın komşuların bulunması için, verilerin indekslenmesi gerekir ve bu süreç genellikle yüksek işlem maliyetlidir. Annoy kütüphanesi ile oluşturulan indeks dosya sistemi üzerine kaydedilebilir. Bu sayede herhangi bir arama işlemi gerçekleştirmek için, dosya sistemi üzerindeki indeks dosyası mmap ile belleğe yüklenir ve anında arama yapılabilir hale gelmektedir.

Annoy kütüphanesi Spotify tarafından, kullanıcılara müzik önerileri yapılırken kullanılmaktadır. Spotify üzerinde her kullanıcı ve öğe çok boyutlu bir uzay üzerinde birer vektör olarak temsil edilmektedir. Annoy kütüphanesi Spotify tarafından, çok boyutlu bir uzay üzerinde benzer kullanıcılar ve öğelerin aranması için kullanılmaktadır.

Annoy, uzaklık hesaplaması için; Öklid mesafesi, Manhattan mesafesi, Kosinüs mesafesi, Hamming mesafesi gibi mesafe hesaplama yöntemlerini kullanabilmektedir.

3.2.2. FALCONN

FALCONN en yakın komşu arama problemi için algoritmalar içeren bir açık kaynak kütüphanedir. FALCONN yüksek boyutlu uzaylarda en yakın komşu araması için popüler bir yöntem olan LSH (Locality-Sensitive Hashing) yöntemini kullanmaktadır (Andoni ve diğ., 2015). FALCONN geliştiricileri amaçlarını, LSH tabanlı veri

yapılarının çok verimli ve iyi test edilmiş bir şekilde uygulanmasını sağlamak olarak tanımlamaktadır.

3.2.3. FLANN

FLANN yüksek boyutlu uzaylarda yaklaşık en yakın komşu aramaları yapmak için geliştirilmiş açık kaynak bir kütüphanedir. En yakın komşu araması için en iyi şekilde çalışan algoritmalar koleksiyonunu ve veri kümesine bağlı olarak en iyi algoritmayı ve optimum parametreleri otomatik olarak seçmen bir sistemi içerir.

Yaklaşık en yakın komşu arama algoritmalarının çoğu pratik uygulamada ve senaryoda çok daha hızlı olduğu kanıtlanmıştır. FLANN (Fast Library for Approximate Nearest Neighbors) C++ programlama dili ile geliştirilmiş, C, MATLAB ve Python ile kullanılabilen bir kütüphanedir (Muja ve Lowe, 2009).

3.2.4. Hnswlib

Hnswlib yaklaşık en yakın komşu araması problemi için C++ ile geliştirilmiş, çizge tabanlı bir kütüphanedir.

Vektör uzayında arama problemleri için NSW (Navigable Small World) çizgeleri kullanılarak arama süresinin kısaltılması hedeflenmektedir (Ponomarenko ve diğ., 2011). NSW çizgelerinde her köşe noktası birkaç başka köşeye bağlanır. Birbirleriyle bağlantı kuran köşeler arkadaş köşesi olarak tanımlanır ve her köşe için arkadaş köşe listeleri tutularak çizge oluşturulur. Bir NSW çizgesi üzerinde arama işlemi yapılırken önceden tanımlanmış bir giriş noktasından arama işlemi başlatılır ve giriş noktasından sorgu vektörüne en yakın olan köşelere geçerek arama işlemi sürdürülür. Algoritma açgözlü bir yaklaşımla sorgu vektörüne daha yakın olan bağlantılı köşelere geçerek arama yapar.

Hnswlib ile çizge oluşturulurken katmanlı bir çizge yapısı oluşturulmakta ve vektörler yinelemeli olarak birer birer çizge'ye eklenmektedir. Bir vektörün belirli bir katmana yerleştirilmesine veya yerleştirilmemesine algoritma tarafından bir olasılık fonksiyonu ile hesaplanarak karar verilmektedir (Malkov ve Yashunin, 2018).

4. METİN ÖZETLEME SİSTEMLERİ

Metin özetleme sistemleri, bilgi kaynaklarından elde edilen metinlerin kullanıcı ya da uygulamalar tarafından ihtiyaç duyulan en önemli kısımlarının elde edilmesini sağlayan sistemlerdir (Mani ve Maybury, 1999).

Metin özetleme sistemleri genel olarak çıkarıma dayalı özetleme ve yoruma dayalı özetleme olmak üzere ikiye ayrılmıştır. Çıkarıma dayalı özetleme, bir belgeden önemli bilgilerin, paragrafların seçilmesi ve özet olacak şekilde yeniden birleştirilmesidir (Gupta ve Lehal, 2010). Çıkarıma dayalı özetleme yalnızca istatistiksel yöntemlere dayanır. Dolayısıyla elde edilen özet, özetlenen metnin kısa bir halidir. Bu sebeple düşük maliyetlidir.

Yorumlayarak özetleme, özetlenecek metinde semantik yaklaşımlarla bir özet çıkarmaya ve sonrasında doğal dil işleme teknikleri ile insan diline yakın bir sonuç üretmeye çalışır. Çıkarıma dayalı özetleme ile yorumlayarak özetlemeyi ayıran en belirgin fark, yorumlayarak özetleme ile elde edilen özetin, özetlenecek metnin yazıldığı dile bağımlı olarak dilbilimsel yöntemlerin kullanılması ve özetin yalnızca metnin kısaltılmış bir hali olmayıp sistemin metni anlayarak yeniden yorumlanmış bir hali olmasıdır. Yorumlayarak özetleme yaklaşımları istatistiksel yöntemlerle birlikte, doğal dil işleme yöntemlerini de etkin olarak kullanmaktadır.

Metin özetleme aşamaları Eduard ve Lin tarafından, konu belirleme, yorumlama ve özet üretme olacak şekilde 3 faz olarak belirlenmiştir (Eduard ve Lin, 1998).

4.1. Çıkarıma Dayalı Özetleme

Bir metnin özetlenme sürecinde çıkarıma dayalı özetleme, istatistiksel yöntemlere dayanır. Cümle seçme, doğal dil işleme teknikleri ve dil bilgisinden bağımsızdır. Bu açıdan düşük maliyetlidir. Cümle seçme, metin özetleme sistemlerinde yalnızca önemli cümlelerin belirli ön kabul ve önsezilerle filtrelenmesi işlemidir. Cümle seçme tekniklerinin en büyük dezavantajı özet metin ile özetlenen metin arasında ortaya çıkabilen tutarsızlıklardır. Buna rağmen cümle seçme teknikleri ile özet metin, özetlenen metin hakkında ipuçları verebilir. Metin özetleme ile ilgili yapılan çalışmaların büyük bir çoğunluğu metindeki önemli cümlelerin seçilmesiyle özet oluşturan özetleme

yaklaşımlarına odaklanmıştır (Kupiec ve diğ., 1995). İstatistiksel yöntemler anahtar kelimeleri ve cümleleri bulmak için sıklıkla kullanılmaktadır. Cümle seçimlerinde başarımı yükseltmek adına farklı öznitelikler, seçilecek cümlelerin değerlendirmesi için kullanılmaktadır. Cümlelerin skorlarının belirlenmesi için, Fattah ve Ren cümlenin konumu, cümledeki negatif ve pozitif kelimeler, diğer cümlelerle benzerlik gibi öznitelikler kullanmıştır (Fattah ve Ren, 2008). Salton ve Buckley ise öznitelik olarak terim frekanslarını kullanmıştır (Salton ve Buckley, 1988). Belirlenen öznitelikler aynı ağırlık değerlerine sahip olabileceği gibi özetlenecek metnin konusuna göre farklılıklar da gösterebilir. Öznitelikler ve her özniteliğin ağırlığı kullanılarak cümleler için skor değeri belirlenmektedir. Metinde yer alan cümleler bu skor değerlerine göre sıralanır ve en yüksek skora sahip cümleler en önemli cümleler olarak değerlendirilir. Her özniteliğe aynı ağırlığın verilmesi durumunda sonuçlar yanıltıcı olabilmektedir. Farklı konulardaki metinler aynı öznitelikler için aynı ağırlığa sahip olmamalıdır.

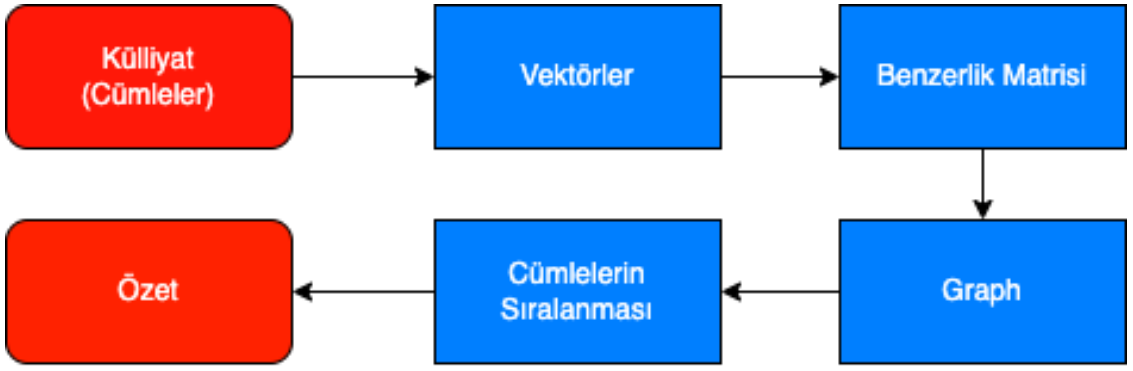
Altan (2004) Türkçe ekonomi makalelerini özetlemek için 19 adımlı bir sistem geliştirmiş ve çok pozitif, pozitif ve negatif kelimeler gibi özniteliklerle cümleleri skorlamıştır. Uzundere ve diğ. (2008) haberlerin özetlenmesinde önemli olduğu düşünülen cümlelerin puanlanması için; cümlenin başlıkta ve varsa alt başlıklarda geçen kelimeleri içerip içermediğini, metin içerisinde yer alan kelimelerin frekanslarını, cümlenin metin içerisindeki konumunu, özet çıkarılırken kullanılmak üzere anahtar kelimeleri, özetlenecek metinde yer alan özel isimleri, pozitif ve negatif kelimeleri, eşdizimli kelimeleri, cümle içerisinde geçen sayıları, alıntı cümlelerin anlaşılabilmesi için çift tırnak işaretini, cümlenin bitiş işareti olarak ünlem ve soru işaretlerini, cümlelerin ortalama uzunluklarını ve cümlelerde geçen gün ay isimlerini öznitelik olarak kullanmış ve her bir öznitelik için ayrı bir ağırlık değeri belirlemiştir. Bu sayede, haber metnindeki cümlelerin başlıktaki kelimeleri içermesi, cümlenin haber metninin giriş cümleleri olması gibi durumların ağırlık değeri yüksek tutularak cümle seçimindeki başarım artırılmıştır.

4.1.1. Textrank Algoritması

Textrank algoritması çizge tabanlı bir metin özetleme algoritmasıdır (Mihalcea ve Tarau, 2004). Textrank algoritması PageRank algoritmasını temel alır. Textrank bir metin içerisinden anahtar sözcükleri ve çizge tabanlı bir modelle kelimelerin önem

ağırlıklarını çıkarır. Cümleler içerisinden anahtar kelime sayısı fazla olanlar yüksek skora sahip olur.

Pagerank algoritmasında bir web sayfasının önemi belirlenirken, web sayfaları arasındaki referans ilişkileri kullanılır. Belirlenen PR değeri ile bir kullanıcının ilgili sayfayı ziyaret etme olasılığı temsil edilir. Textrank algoritmasında ise web sayfaları arasındaki referans ilişkileri bir metni oluşturan cümleler arasında kurulmuştur. Cümleler arasındaki benzerlikler, web sayfaları arasında kurulan referans ilişkisine benzetilebilir. TextRank algoritması akış diyagramı Şekil 4.1.'de gösterilmektedir.



Şekil 4.1. TextRank algoritması akış diyagramı

Textrank algoritmasında, özetlenecek metin ön işlemlerden geçirildikten sonra cümlelere bölünür. Cümleler üzerinde oluşturulan dil modeli ile cümle vektörleri oluşturulacak benzerlik matrisi elde edilir. Benzerlik matrisi bir çizge üzerinde tutularak, benzerlik skorlarına göre cümleler seçilerek özet metin elde edilir. Çizge üzerinde her bir node bir cümleyi, edgeler ise metni oluşturan cümleler arasındaki benzerliği temsil eder.

4.1.2. Lexrank Algoritması

Lexrank algoritması, Textrank algoritması gibi çizge tabanlı bir metin özetleme algoritmasıdır. Lexrank, Pagerank algoritmasını temel alır. Çıkarıma dayalı özetleme, orijinal belgelerdeki cümlelerin bir alt kümesini seçerek çalışır. Bu süreç, kümenin ana temasıyla ilgili gerekli ve yeterli miktarda bilgiyi veren (çok belgeli) bir kümedeki en merkezi cümlelerin belirlenmesi olarak görülebilir. Bir cümlenin merkeziliği, genellikle içerdiği kelimelerin merkeziliği açısından tanımlanır. Sözcük merkeziliğini değerlendirmenin yaygın bir yolu, bir vektör uzayında belge kümesinin merkezine

bakmaktır. Bir kümenin merkezi, önceden tanımlanmış bir eşiğin üzerinde TF-IDF puanlarına sahip kelimelerden oluşan bir sözde belgedir; burada tf, kümedeki bir kelimenin sıklığıdır ve idf değerleri tipik olarak çok daha büyük ve benzer bir tür veri seti üzerinden hesaplanır (Erkan ve Radev, 2004).

Lexrank algoritmasında, özet metnin oluşturulmasında kullanılacak cümlelerin seçilmesinde kriter olarak diğer cümlelerle benzerliği temel alınır. Metni oluşturan cümlelerden seçilerek özet metinde kullanılacak bir cümlenin özetlenen metinde bulunan diğer bazı cümlelerle benzer olması gerekmektedir. Lexrank algoritması çizge üzerinde tutulan benzerlik oranlarının hesaplanması noktasında Textrank algoritmasından farklılaşmaktadır. Textrank algoritmasında her bir kelimenin ağırlığı kullanılarak cümlelerin önemi belirlenirken, Lexrank algoritması ağırlıkların hesaplanması için cümlelerin merkeziliğini kullanmaktadır.

Lexrank algoritmasında cümleler arasındaki benzerlikler belirlenirken, denklem (4.1)'deki modifiye edilmiş bir cosine formülü kullanılmaktadır.

$$\text{idf-modified-cosine}(x, y) = \frac{\sum_{w \in x,y} \text{tf}_{w,x}\text{tf}_{w,y}(\text{idf}_w)^2}{\sqrt{\sum_{x_i \in x}(\text{tf}_{x_i,x}\text{idf}_{x_i})^2} \times \sqrt{\sum_{y_i \in y}(\text{tf}_{y_i,y}\text{idf}_{y_i})^2}} \tag{4.1}$$

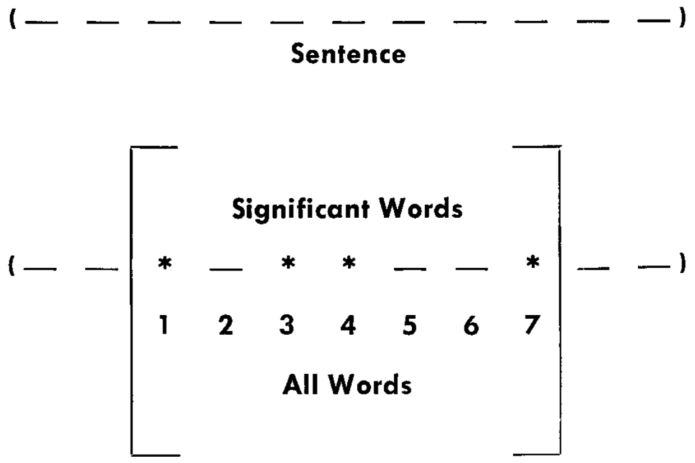
Lexrank algoritmasında, özetlenecek metin bir ön işleme sürecinden geçirilir. Metni oluşturan cümleler ayrıştırılır. Oluşturulan dil modeli kullanılarak cümle vektörleri bir çizgeye (graph) çevrilir. Çizge üzerindeki her bir node bir cümleyi, edgeler ise cümleler arasındaki benzerliği temsil eder. Cümleler arasındaki benzerlikler modifiye edilmiş bir idf cosine formülü ile hesaplanır. Formül ile iki cümle arasındaki mesafe hesaplanır. Oluşturulan benzerlik matrisi üzerinden cümleler sıralanarak, özet metinde kullanılacak cümleler seçilir ve özet metin elde edilir.

4.1.3. Luhn Algoritması

Metin özetleme alanındaki çalışmalar erken bilgisayar dönemlerinde başlamıştır. Mikroçiplerin henüz bilgisayarlarda kullanılmadığı bir dönemde, 1958 yılında Hans Peter Luhn tarafından yayınlanan çalışma ile otomatik bir metin özetleyici

geliştirilmiştir. Luhn geliştirdiği sezgisel algoritma ile, IBM 704 veri işleme makinesi üzerinde makalelerin otomatik olarak özetlerinin çıkarılmasını sağlamıştır.

Luhn, bir makalenin yazılması sürecinde yazar tarafından konunun detaylandırılması için bazı kelimelerin sıkça kullanıldığını ve bu kelimelerin makale için önemli olabileceğini savunur. Algoritma özetlenecek metin içerisinde kelime frekanslarını kullanarak anahtar kelimeleri belirler. Metin içerisinde geçen kelimelerin sıklığını kullanarak en fazla sayıda ve en az sayıda geçen kelimeler belirlenir. Özetlenecek metin içerisindeki en fazla ve en az sayıda bulunan kelimeler değerlendirme dışında bırakılır. Luhn algoritmasında, metni oluşturan cümlelerin ağırlıkları hesaplanırken pencereleme yapılır. Cümleler içerisindeki önemli kelimelerin geçtiği bölümler pencerelenir. Şekil 4.2.'de önem faktörünün hesaplanması gösterilmektedir. Cümlenin ağırlığı için önem faktörü tanımı yapılmıştır. Önem faktörü, cümle içerisinde pencereleme yapılan cümle bölümünde geçen önemli kelimelerin toplam sayısının karesinin, pencereleme yapılan cümle bölümünde geçen toplam kelime sayısına bölümüyle hesaplanır.



Şekil 4.2. Luhn algoritmasında cümlenin önem faktörünün hesaplanması (Luhn, 1958)

Pencere içerisindeki önemli kelime sayısının karesi: 4^2 = 16

Pencere içerisindeki toplam kelime sayısı: 7

Önem faktörü: 16 / 7 = 2.3

Luhn algoritmasında, özetlenecek metnin cümleleri önem faktörüne göre sıralanır ve özet metni oluşturacak olan cümleler seçilerek özet metin elde edilir.

4.1.4. Latent Semantic Analysis (LSA)

Gizli anlam analizi / Latent Semantic Analysis (LSA) bir doküman üzerinde yapılan istatistiksel hesaplamalarla dokümanı oluşturan kelimelerin bağlamsal olarak kullanım anlamlarını çıkarmak ve kelimelerin bağlamsal kullanımlarını temsil etmek için önerilen bir yöntemdir (Landauer ve Dumais, 1997). Gizli anlam analizi doğal dil işlemede dokümanları ve dokümanları oluşturan terimleri kullanarak doküman ve içerdiği terimler arasındaki ilişkiyi analiz eder.

LSA bir dokümandaki bilgilerin aranması, dokümanları tanımlayan üst verilerin elde edilmesi gibi amaçlarla kullanıldığı durumlarda gizli anlam indeksleme / Latent Semantic Indexing (LSI) olarak da adlandırılmaktadır. Gizli anlam analizi konu belirleme, metin özetleme, metin sınıflandırma gibi problemlerde kendisine kullanım alanı bulmaktadır.

Gizli anlam analizinde bir metni oluşturan kelimelerin frekansları kullanılarak istatistiksel analizle bir anlamsal uzay oluşturulur. Gizli anlam analizinde anlamsal uzay oluşturulan bu süreç eğitim sürecidir. Metni oluşturan kelimelerin anlamsal özellikleri bir dizi vektörle ifade edilir. Bu vektörler anlamsal vektör olarak adlandırılmaktadır. Gizli anlam analizindeki temel varsayım, benzer anlamlardaki kelimelerin metnin benzer bir bölümünde yer alacağıdır.

Gizli anlam analizinin insan bilgisini yansıtması yönünden yeterliliği çeşitli şekillerde belirlenmiştir. LSA tarafından üretilen kelime ve paragraf anlam temsillerinin, insan bilişsel fenomenlerini simüle etme yeteneğine sahip olduğu bulunmuştur. Kelime anlamlarının karakterizasyonu için gizli anlam analizi pratik bir yöntemdir. Gizli anlam analizinin kelime-kelime, kelime-paragraf ve paragraf-paragraf ilişkileri insan bilişsel fenomenlerinden çağrışım veya anlamsal benzerlik kurması gibi yetenekleriyle bağlantılıdır (Dumais, 2004).

Tablo 4.1. Örnek bir doküman listesi

Doküman-1	Uludağ kar kış bot.
Doküman-2	Kar kış mont.
Doküman-3	Uludağ kış eldiven.

Tablo 4.2. Terim-doküman matrisi

Kelimeler	Doküman-1	Doküman-2	Doküman-3
uludağ	1	0	1
kar	1	1	0
kış	1	1	1
bot	1	0	0
mont	0	1	0
eldiven	0	0	1

Gizli anlam analizinde bir belgeyi oluşturan terimler girdi olarak ele alınır. Tablo 4.1.'de örnek bir doküman listesi gösterilmektedir. Belgeyi oluşturan paragraflar üzerinde terim-doküman matrisi oluşturulur. Oluşturulan matriste satırlar her bir paragrafta bulunan benzersiz kelimeleri, sütunlar ise her bir paragrafı temsil eder. Bir belge üzerinde çok sayıda geçen ancak anlamsal olarak fazla etkisi olmayan kelimeler olabilir. Bu kelimeler durak kelimeler, kısaltmalar, zamirler olabilir. Bu sebeple, terim-doküman matrisinde yer alan yüksek frekanslı ancak gereksiz olduğu değerlendirilen kelimelerin ağırlıklarının azaltılması amacıyla TF-IDF dönüşümü kullanılmaktadır. Tablo 4.2'de ise Tablo 4.1.'deki örnek dokümanlar kullanılarak elde edilen terim-doküman matrisi gösterilmektedir. Oluşturulan birinci seviye terim-doküman matrisi üzerinde bir indirgeme yapılır. Gizli anlam analizinde tekil değer ayrıştırması / Singular Value Decomposition (SVD) yapılarak dokümandaki gizli konuları elde edilir. Tekil değer ayrıştırması temel olarak matrisin satır sayısını azaltma işlemidir. Tekil değer ayrıştırması yapılırken matriste yer alan satır sayısı yani kelime sayısı azaltılırken, sütunlar arasında yani metni oluşturan paragraflar arasındaki benzerlikler korunur.

4.2. Yoruma Dayalı Özetleme

Yoruma dayalı özetleme, özetlenecek metinden cümle seçilmesi ve seçilen cümlelerin de yeniden oluşturularak kısaltılmasını amaçlayan metin özetleme yöntemidir. Yoruma dayalı özetleme, ana metnin temel kavramlarını ve ilgili bilgileri anlamayı ve daha sonra bu bilgiyi kısa ve net bir biçimde ifade etmeyi içermektedir. Yoruma dayalı özetleme yaklaşımları istatistiksel yöntemlerle birlikte, doğal dil işleme yöntemlerini de

etkin olarak kullanmaktadır. Yoruma dayalı özetleme teknikleri genel olarak yapısal temelli ve semantik temelli yaklaşım olmak üzere iki kategoriye ayrılır (Khan, 2014).

Yapısal temelli yaklaşım, şablonlar, çıkarma kuralları ve ağaç yapısını kullanarak metinlerdeki en önemli bilgiyi şemalar yoluyla kodlar (Genest ve Lapalme, 2011). Bu yaklaşımı kullanan farklı yöntemler mevcuttur. Bunlar; ağaç tabanlı yöntem, şablon tabanlı yöntem, ontoloji tabanlı yöntem, öncü ve temel ifade tabanlı yöntem ve kural tabanlı yöntemdir. Semantik temelli yöntemde, doğal dil üretim sistemini beslemek için belgelerin semantik gösteriminden faydalanılır. Bu yöntem, dilsel verileri işleyerek isim cümlelerini ve fiil cümlelerini tanımlamaya odaklanmaktadır (Saggion ve Lapalme, 2002). Semantik temelli yaklaşımı kullanan yöntemler; çok modlu semantik model, bilgiye dayalı yöntem ve semantik çizge tabanlı yöntemdir. Yoruma dayalı özetleme ile elde edilen özet yalnızca ana metnin kısaltılmış bir hali değildir. Özet metin, dilbilimsel yöntemlerle incelenir. Cümleler doğal dil işleme teknikleri ile daha kısa ve net bir biçimde oluşturulur.

Haber metinlerinin yoruma dayalı özetleme yöntemleri ile özetlenmesi beraberinde birtakım problemler getirmektedir. Alıntı ifadelerin net bir şekilde tespit edilmesi, neden sonuç çerçevesinde yazılmış cümlelerin anlam bütünlüğünün korunması gerekmektedir. Pratikte haber metinlerinin dil bilgisi kurallarına özen gösterilmeden oluşturulması, başlık ve içerik arasındaki tutarsızlıklar gibi sebeplerle yoruma dayalı özetleme yöntemlerinin uygulanması daha da zorlaşmaktadır. Ayrıca, dillerin kendilerine özgü yapıları ve kuralları sebebiyle sistemlerin ilgili dile özel geliştirilmesi gerekmektedir.

4.3. Özetleme Performans Değerlendirme Metrikleri

4.3.1. Rouge

ROUGE temel olarak, otomatik metin özetleyici bir sistem tarafından oluşturulan otomatik özetleri veya makine tarafından yapılmış bir makine çevirisi performansını değerlendirmek için bir dizi metrikten oluşan bir performans ölçme yöntemidir. Metin özetleme alanında; otomatik olarak üretilen bir özeti, tipik olarak insan tarafından üretilen bir dizi referans özetiyle karşılaştırarak çalışır. ROUGE terimi, Recall-Oriented Understudy for Gisting Evaluation'ın kısaltmasıdır. Ölçüler, bilgisayar tarafından

oluşturulan özet ile insanlar tarafından oluşturulan ideal özetler arasındaki n-gram, kelime dizileri ve kelime çiftleri gibi örtüşen birimlerin sayısını saymaktadır (Lin, 2004). Özetleme performansı değerlendirilirken, sistem tarafından oluşturulmuş özet ile tipik olarak insan tarafından oluşturulan özet karşılaştırılmaktadır.

Tablo 4.3. Makine özeti ve referans özet

Makine Özeti	NASA yakında devasa büyük bir gezegen açıklayacak
Referans Özet	NASA yakında büyük bir gezegen açıklayacak

Örneğin Tablo 4.3.'deki gibi makine özeti ve insan tarafından oluşturulmuş referans özet ele alınacak olursa, ayrı ayrı kelimeler değerlendirildiğinde iki özet arasında toplam 6 adet örtüşen kelime bulunmaktadır. Ancak bu değer, özetleme performansı açısından tek başına bir anlam ifade etmeyecektir. Daha anlamlı bir değer elde etmek için, iki özet arasındaki örtüşmeyi kullanarak precision(hassasiyet) ve recall(kesinlik) değerleri hesaplanmalıdır.

Metin özetleme performansı açısından ele alınacak olursa recall makine özeti tarafından referans özetin ne kadarının kurtarıldığını ifade eder. Teker teker kelimeler ele alındığında recall denklem (4.2)'de gösterildiği şekilde, precision ise denklem (4.3)'de gösterildiği şekilde hesaplanmaktadır.

$$r = toplam \text{ örtüşen } kelime/referans \text{ özetteki } toplam\ kelime \quad (4.2)$$

$$p = toplam \text{ örtüşen } kelime/makine \text{ özetindeki } toplam\ kelime \quad (4.3)$$

Tablo 4.3.'deki örneğe göre recall değeri; recall = 6 / 6 = 1 olarak hesaplanacaktır. Yalnıza recall değeri düşünülecek olursa bu referans özetteki tüm kelimelerin makine özeti tarafından yakalandığı anlamına gelecektir.

Tablo 4.3.'deki örneğe göre precision değeri; precision = 6 / 7 = 0.86 olarak hesaplanacaktır. Yani makine özetindeki 7 kelimeden 6'sının aslında ilgili veya gerekli olduğu anlamına gelmektedir.

Yalnızca recall ve precision hesaplamaları, özetleme performansı hakkında her zaman bir anlam ifade etmemektedir. Örneğin; makine özetinin referans özetten çok daha fazla

sayıda kelime içerdiği durumlarda precision düşecek recall artacaktır. Bu sebeple precision ve recall değerlerinin harmonik ortalamasının alınmasıyla elde edilen f-score(f-puanı) hesaplanmaktadır. f-score denklem (4.4)'de gösterildiği şekilde hesaplanmaktadır. Tablo 4.3.'deki örneğe göre f-score değeri; f-score = 2 * (0.86 * 1) / (0.86 + 1) = 0.92 olarak hesaplanacaktır.

$$f - score = 2 * (p * r) / (p + r) \quad (4.4)$$

LCS(Longest Common Subsequence) / En uzun ortak küme, iki küme arasındaki en uzun ve sıralı bir şekilde bulunan ortak elemanları ifade eden alt kümedir. Örneğin; K1 kümesi için K1-> {nasa, yakında, devasa, büyük, bir, gezegen, açıklayacak} kümesi ile, K2 kümesi için K2-> {nasa, yakında, büyük, bir, gezegen, açıklayacak} kümesi arasındaki, en uzun alt küme LCS->{büyük, bir, gezegen, açıklayacak} olarak bulunacaktır.

ROUGE-N: Makine tarafından üretilen özet ile referans özet arasındaki unigram(n=1), bigram(n=2), trigram(n=3) ve daha yüksek dereceli n-gram örtüşmesini ölçer. Örneğin ROUGE-1, makine özeti ile referans özet arasındaki unigram yani her kelimenin örtüşmesini ifade etmektedir. ROUGE-2 makine özeti ile referans özet arasındaki bigramların örtüşmesini ifade etmektedir.

ROUGE-L: En uzun ortak küme (LCS) kullanarak makine özeti ile referans özet arasındaki eşleşen en uzun kelime dizisini ölçer. ROUGE-L ile iki özet arasındaki en uzun sıralı ortak n-gramler aranacağı için önceden tanımlanmış bir n-gram uzunluğuna ihtiyaç duyulmamaktadır.

ROUGE-W: Ardışık en uzun ortak kümeleri (LCS) kullanarak, ağırlıklı LCS tabanlı bir yöntemle özetleyicinin performansını ölçer.

ROUGE-S: Bir cümlede rastgele boşluklara izin veren herhangi bir kelime çiftidir. ROUGE-S aynı zamanda atlama gramı (skip-gram) olarak da adlandırılmaktadır. Örnek olarak skip-bigram kelimeler arasında en fazla iki boşluk olabilen kelime çiftlerinin örtüşmesini ölçecektir. Örneğin; "istanbul bugün biraz yorgun" ifadesinin skip-bigramleri "istanbul bugün, istanbul biraz, istanbul yorgun, bugün biraz, bugün yorgun, biraz yorgun" olacaktır. Tablo 4.3.'deki örnek üzerinden ROUGE-2 precision

ve recall hesaplamaları yapılacak olursa makine özeti bigramları; {NASA yakında, yakında devasa, devasa büyük, büyük bir, bir gezegen, gezegen açıklayacak}, referans özet bigramları ise; {NASA yakında, yakında büyük, büyük bir, bir gezegen, gezegen açıklayacak} şeklinde olacaktır. Bu bigramlar üzerinden ROUGE-2 recall değeri 4/5 = 0.8 olarak hesaplanacaktır. Precision değeri ise 4/6 = 0.67 olarak hesaplanacaktır. Buradaki precision değeri makine özeti bigramlerinin referans özetiyle %67'lik bir örtüşme sağladığını ifade etmektedir.

4.3.2. Bleu

BLEU - Bilingual Evaluation Understudy (İki Dilli Değerlendirme Altyapısı) makine çevirilerinin performansını ölçmek için geliştirilmiş bir yöntemdir. İlk olarak makine çevirileri için geliştirilmiş olmasına rağmen, özetleme gibi doğal dil işleme görevlerinde de kullanılmaktadır.

BLEU, makine tarafından bir dilden bir başka dile çevirisi yapılmış makine çevirisini, insan tarafından yapılmış referans çeviriyle karşılaştırır. BLEU Kishore Papineni vd. tarafından 2002 yılında önerilmiştir. BLEU skoru hesaplanırken geliştirilen yaklaşım, makine çevirisindeki n-gramler ve referans çevirideki n-gramler arasından eşleşenlerin sayılmasıdır. Yaklaşımda unigramler her bir kelimeyi, bigramler ise her bir kelime çiftini ifade etmektedir. Karşılaştırma yapılan metinlerdeki kelime sırasına bakılmaksızın karşılaştırma yapılır. Bir BLEU uygulayıcısı için birincil programlama görevi, adayın n-gramlarını referans çevirisinin n-gramları ile karşılaştırmak ve eşleşme sayısını saymaktır. Bu eşleşmeler pozisyondan bağımsızdır. Ne kadar çok eşleşme olursa, aday çevirisi o kadar iyi olur (Papineni ve diğ., 2002).

BLEU skoru 0 ile 1 arasında bir değer alır. 1 makine çevirisi ile referans çevirinin mükemmel bir eşleşme sağladığını, 0 ise makine çevirisi ile referans çevirinin hiç eşleşmediğini ifade eder.

5. DİL MODELİ GELİŞTİRME

5.1. Bag of Words (BOW)

Bag of Words modeli, doğal dil işlemede ve makine öğrenmesinde bir metnin belirli bir kelime haznesi(vocabulary) üzerinde temsil edilmesini sağlayan bir dil modelidir. Bag of Words modeli ile metin, makine öğrenmesi algoritmalarında kullanılmak üzere vektörlerle ifade edilebilmektedir. Bag of Words modeli doğal dil işleme görevlerinde, doküman ve metin sınıflandırma algoritmalarında yaygın olarak kullanılmaktadır.

İki cümleden oluşan bir metnin bag of words modeli ile modellenmesi için aşağıdaki adımlar uygulanır.

Cümle-1: İstanbul yağmur yağdığında güzeldir. Ankara da karlı güzeldir.

Cümle-2: İstanbul dolu yağdığında da güzeldir.

İki cümleden elde edilen kelime listesi şu şekilde olacaktır;

[“istanbul”, ”yağmur”, ”dolu”, ”yağdığında”, ”güzeldir”, ”ankara”, ”karlı”, ”da”]

Cümlelerden elde edilen kelime listesinin sıklığına göre sıralanması durumunda elde edilecek tablo, Tablo 5.1.’de gösterildiği şekilde olacaktır.

Tablo 5.1. Kelime sıklık tablosu

Kelime	**güzeldir**	**istanbul**	**yağdığında**	**da**	**ankara**	**karlı**	**yağmur**	**dolu**
Sayı	3	2	2	2	1	1	1	1

Cümlelerde geçen kelimelerin geçme sıklığına göre sıralanması durumunda, en sık kullanılan 5 kelimenin bir matris üzerinde gösterimi ise Tablo 5.2.’de gösterildiği şekilde olacaktır.

Tablo 5.2. Bag of Words modeli

	güzeldir	**istanbul**	**yağdığında**	**da**	**ankara**
Cümle-1	1	1	1	1	0
Cümle-2	1	1	1	1	1

Oluşturulan matris örnekteki iki cümleden oluşan bir metni temsil eden bag of word modeli olacaktır. Matristeki 0 değerleri ilgili kelimenin cümlede yer almadığı, 1 değeri ise ilgili kelimenin cümlede yer aldığını temsil etmektedir. Bag of Words modeli ile metinde yer alan kelimelerle ilgili bir sıklık değeri saklanmamaktadır. Bag of Words modelinde tüm kelimeler aynı önem değerine sahiptir.

5.2. TF-IDF

TF-IDF, bir dokümanı oluşturan terimlerin doküman içerisindeki önemini gösteren ağırlık faktörüdür. Term frequency (TF) yani terim frekansı bir kelimenin geçtiği cümle içerisindeki frekansını temsil eder. Inverse document frequency (IDF) yani ters doküman frekansı ise bir kelimenin tüm doküman üzerindeki değerini tanımlar. TF ve IDF değerlerinin çarpımı ise TF-IDF değerini verir.

Örnek olarak iki cümleden oluşan bir metnin TF-IDF matrisi aşağıdaki şekilde hesaplanmaktadır.

Cümle-1 : bugün güzel bir şey olacak

Cümle-2 : yarın bugünden güzel olacak

TF değerinin hesaplanması için aşağıdaki formül kullanılmaktadır.

TF = (Bir kelimenin cümle içindeki geçme sayısı) / (Cümle içindeki toplam kelime sayısı)

Tablo 5.3.'te örnek cümlelerden TF dönüşümü ile elde edilen, kelimelerin terim frekansları gösterilmektedir.

Tablo 5.3. Örnek cümlelerden TF dönüşümü

	bugün	**güzel**	**bir**	**şey**	**olacak**	**yarın**	**bugünden**
C-1	0.2	0.2	0.2	0.2	0.2	0	0
C-2	0	0.2	0	0	0.2	0.2	0.2

IDF dönüşümü için denklem (5.1) kullanılmaktadır.

$$IDF = log(Cümle\ Says\ /\ Kelimenin\ geçtiği\ cümle\ says) \quad (5.1)$$

Tablo 5.4. Örnek cümlelerden IDF dönüşümü

	bugün	güzel	bir	şey	olacak	yarın	bugünden
IDF	0.301	0	0.301	0.301	0	0.301	0.301

TF-IDF hesaplaması için, metni oluşturan cümlelerden elde edilen kelimelerin TF ve IDF değerlerinin hesaplanmasının ardından, her bir kelime için bu değerler çarpılarak TF-IDF dönüşümü hesaplanmaktadır. Tablo 5.4.'te örnek cümlelerden IDF dönüşümü, Tablo 5.5.'te ise TF-IDF dönüşümü sonucunda elde edilen değerler gösterilmektedir.

Tablo 5.5. Örnek cümlelerden TF-IDF dönüşümü

	bugün	güzel	bir	şey	olacak	yarın	bugünden
C-1	0.0602	0	0.0602	0.0602	0	0.0602	0
C-2	0	0.0602	0	0	0	0.0602	0.0602

5.3. N-Gram

N-gram kendisinden daha uzun bir karakter dizisinin N karakterli bölümünü ifade eder. N-gram ardı ardına gelen karakterlerden oluşmak zorunda değildir. N-gram, bir karakter dizisinin n ve n+i indisindeki karakterlerden oluşan bir N-gram olabileceği gibi, bir karakter dizisinde birlikte meydana gelen herhangi bir karakter kümesi kavramını içerebilir. Otomatik metin özetleme sistemlerinde ise N-gram'ler bir metni oluşturan cümleler veya cümleleri oluşturan kelimeleri kullanmaktadır.

Örnek olarak; "İstanbul'da bugün hava kaç derece olacak?" cümlesini ele alırsak, N=1 olduğu durumda model şu şekilde olacaktır:

"İstanbul'da", "bugün", "hava", "kaç", "derece", "olacak?"

N=2 olduğu durumda model şu şekilde olacaktır:

"İstanbul'da bugün", "bugün hava", "hava kaç", "kaç derece", "derece olacak?"

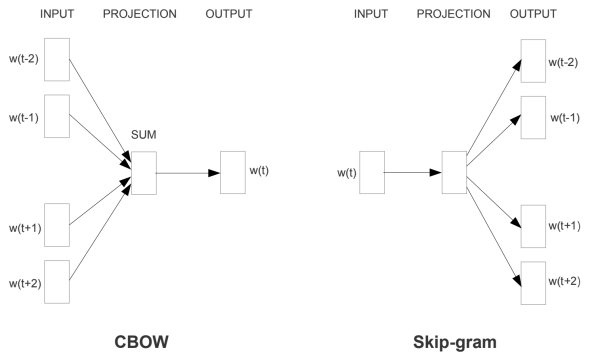
N=3 olduğu durumda ise model şu şekilde olacaktır:

"İstanbul'da bugün hava", "bugün hava kaç", "hava kaç derece", "kaç derece olacak?"

N-gram modelinin örnek kullanım alanlarından birisi olarak, arama motorlarının kelime ve cümle tamamlama özellikleri gösterilebilir. Yine, akıllı asistan uygulamaları, otomatik metin üreten sistemler N-gram modelinin uygulama alanlarındandır.

5.4. Word2Vec

Word2Vec büyük bir metin külliyatını (corpus) kullanarak, kelimelerin ilişkilerini belirlemek için geliştirilmiş tahmin tabanlı bir kelime gömme/temsil (word embedding) yöntemidir. Word2Vec kelimelerin dilsel bağlamlarını oluşturmak için iki katmanlı sinir ağlarını kullanmaktadır. Girdi olarak büyük bir metin külliyatını alarak çıktı olarak bir vektör uzayı oluşturur. Word2Vec kelime temsil modelinde benzer sözcükler birbirine yakın olarak değerlendirilmektedir. Örneğin mandalina ve armut gibi meyve isimlerinin birbirine yakın olması beklenirken, otel ve uzay gemisi kelime çiftinin birbirinden daha uzakta olması beklenmektedir. Word2Vec ile kelime vektörleri oluşturulurken kelimelerin sağında ve solunda kaç kelime olması gerektiğini belirten ve her bir kelimenin kaç boyutlu bir vektör olarak temsil edileceğini belirten hiper parametreler kullanılmaktadır. Vektörler arasındaki kosinüs benzerliği hesaplanarak, bu vektörler tarafından temsil edilen kelimeler arasındaki anlamsal benzerlikler ifade edilmektedir.



Şekil 5.1. Word2Vec mimarisi (Mikolov ve diğ., 2013)

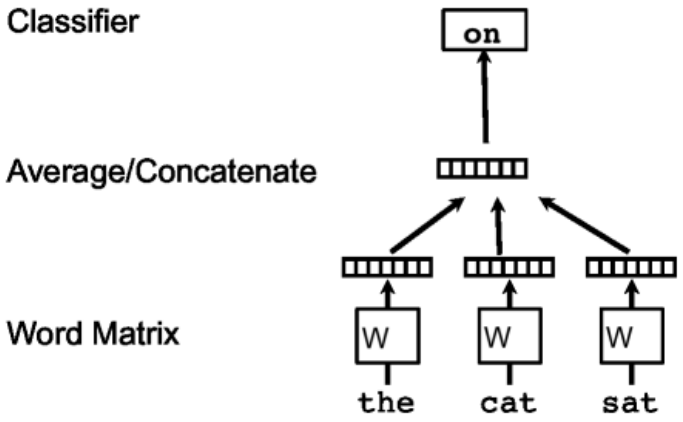
Şekil 5.1.'de Word2Vec mimarisi gösterilmektedir. Word2Vec ile kelimelerin temsili için iki mimari kullanılmaktadır. Bu mimariler CBOW(Continuous Bag of Words) ve Skip-gram mimarisidir. CBOW mimarisinde mevcut kelimeyi çevreleyen kelimelerin bir penceresinden tahmin yapar. Dolayısıyla kelimelerin sırası tahmini etkilemez. Skip-gram mimarisinde ise model, kelimelerin öncesindeki veya sonrasındaki kelimeleri

tahmin etmek için mevcut kelimeyi kullanır. Skip-gram mimarisinde bir kelimenin yakınındaki kelimeler daha uzaktaki kelimelere göre daha fazla ağırlıklandırılır (Mikolov ve diğ., 2013).

5.5. Doc2Vec

Birçok makine öğrenmesi algoritması, girdinin sabit uzunlukta bir özellik vektörü olarak temsil edilmesini gerektirmektedir. Metin verileri ile çalışılması durumunda, en yaygın kullanılan sabit uzunluklu yöntem olarak bag of words gibi yöntemler öne çıkmaktadır. Bag of words yöntemi popüler olmasına rağmen bazı zayıflıklara sahiptir. Örneğin kelimelerin sırasını tutamaz ve kelimelerin anlamlarını görmezden gelir. Bag of words yönteminde güçlü kelimesi, kuvvetli kelimesi ve Paris kelimesi ile aynı uzaklıktadır.

Bag of words yönteminin zayıf yönlerini gidermek amacıyla Word2Vec ile tek tek kelimelerin temsil edildiği yönteme ek bir uzantı olarak, cümleler, paragraflar ve belgeler gibi değişken uzunluktaki metin parçalarından sabit uzunluklu özellik vektörlerinin oluşturulmasını sağlayan gözetimsiz bir algoritma olarak Paragraf vektör modeli önerilmiştir. Bu model ile, kelimelerin sırası ve anlamlarının kaybedilmeden kullanılmasının sağlanması hedeflenmiştir. Paragraf vektör modeli, belgedeki kelimelerin tahmin edilmesi için eğitilmiş birer vektörle belgelerin temsil edilmesini sağlamaktadır. Paragraf vektör modeli herhangi bir uzunluktaki metinlere uygulanabilmektedir.

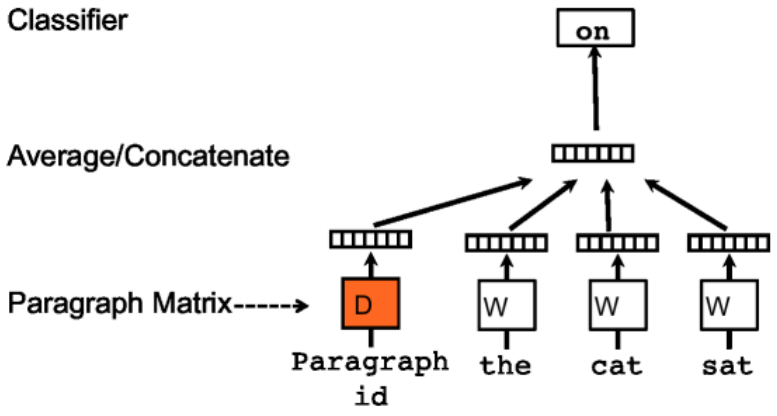


Şekil 5.2. Kelime temsil vektörlerinin oluşturulması (Le ve Mikolov, 2014)

Paragraf vektör modeli Word2Vec modelinin genişletilmiş bir türüdür. Kelime vektörlerinin oluşturulması sürecinde yaygın olarak kullanılan öğrenme modeli şekil 5.2.'de gösterilmektedir. Kelime temsil vektörlerinin oluşturulması için amaçlanan hedef, bir bağlam içerisinde yer alan kelimeler kullanılarak bir kelimenin tahmin edilmesidir. Bu çerçevede, her kelime bir sütunla temsil edilen bir vektörle eşlenmektedir. Sütun ise kelimenin tüm külliyat içerisindeki konumuna göre indekslenir. Vektörlerin birleştirilmesiyle veya toplanmasıyla elde edilen değerler cümledeki sonraki kelimenin tahmin edilmesi için kullanılmaktadır. Şekil 5.2.'deki örnekte, "the", "cat" ve "sat" kelimeleri dördüncü kelime olarak "on" kelimesini tahmin etmek için kullanılmaktadır (Le ve Mikolov, 2014).

Paragraf temsil vektörlerinin öğrenme yaklaşımında kelime temsil vektörlerinin öğrenme yöntemlerinden esinlenilmiştir. Word2Vec modelinde cümledeki bir sonraki kelime hakkında bir tahmin yapılmaktadır. Bu sebeple kelime vektörleri rastgele bir şekilde başlatılmasına rağmen model eğitilirken bir noktada kelimenin anlamına dair bilgiler elde edilmektedir. Paragraf vektörlerinde de benzer bir şekilde paragraf içinde yer alan kelimeler bir sonraki kelimenin tahmin görevine katkıda bulunmaktadır.

Paragraf vektör modelinin eğitilmesine dair model Şekil 5.3.'de gösterilmektedir. Paragraf vektör modelinde her paragraf D matrisinde bir sütunla temsil edilen benzersiz bir vektöre eşlenir ve her kelime ayrıca W matrisinde bir sütunla temsil edilen benzersiz bir vektöre eşlenir. Paragraf vektörü ve kelime vektörlerinin ortalaması alınır ve bir bağlam içerisindeki bir sonraki kelimeyi tahmin etmek için birleştirilir.



Şekil 5.3. Paragraf vektörlerinin oluşturulması (Le ve Mikolov, 2014)

Paragraf vektörlerinin oluşturulması sürecinde eğitim aşamasından sonra paragraf vektörleri paragrafların temsili için kullanılmaktadır.

5.6. fastText

Fasttext, Word2Vec gibi bir metin külliyatını (corpus) kullanarak kelimelerin vektörlerle temsil edilmesini sağlayan ve bunun için yapay sinir ağlarını kullanan bir kelime temsil yöntemidir. 2016 yılında Facebook tarafından geliştirilmiştir.

Büyük metin külliyatlarındaki etiketlenmemiş cümleler kullanılarak oluşturulan kelime temsilleri, birçok doğal dil işleme görevi için kullanışlıdır. Bu tür temsilleri öğrenen popüler modeller, her bir kelimeye ayrı birer vektör atar. Ancak bu durum kelimelerin morfolojisinin görmezden gelinmesine sebep olur. Bu, özellikle geniş kelime dağarcığına ve çok sayıda nadir kelimeye sahip diller için bir sınırlamadır. Fasttext skipgramleri kullanan, her bir kelimenin n-gramleriyle ifade edildiği bir yaklaşımdır. Fasttext her bir kelimeyi sinir ağında teker teker girdi olarak kullanmak yerine, kelimeleri n-gramler olarak parçalar ve girdi olarak kullanır. Bir kelimeyi temsil eden vektör ise bu n-gram vektörlerin toplamı olarak ifade edilmektedir (Bojanowski ve diğ., 2017).

Fasttext ile kelimeler n-gramler olarak ifade edilirken kelimelerin başına ve sonuna “<” ve “>” sembolleri eklenmektedir. Her bir kelimenin temsili için kelimenin n-gramlerine ek olarak kelimenin kendisi de n-gram setine eklenmektedir. Örnek olarak “hakan” kelimesinin n=3 için n-gramleri şu şekilde olacaktır;

<ha, hak, aka, kan, an>

ve kelimenin kendisini ifade eden özel bir sekansı ise şu şekilde olacaktır;

<hakan>

5.7. GloVe

GloVe(Global Vectors), kelimelerin bir vektör uzayında temsil edilmesini sağlamayı amaçlayan gözetimsiz bir öğrenme algoritmasıdır. GloVe literatürdeki iki ana model olan global matris çarpanlara ayırma (global matrix factorization) ve lokal bağlam

penceresi(local context window) modellerinin avantajlarını birleştirerek global log-bilineer yöntemini kullanmaktadır (Pennington ve diğ., 2014). GloVe bir metin külliyatı içerisinden kelime vektörlerini oluşturmak için hem külliyatın tümündeki istatistikleri hem de kelimenin geçtiği daha lokal düzeylerdeki istatistikleri kullanır.

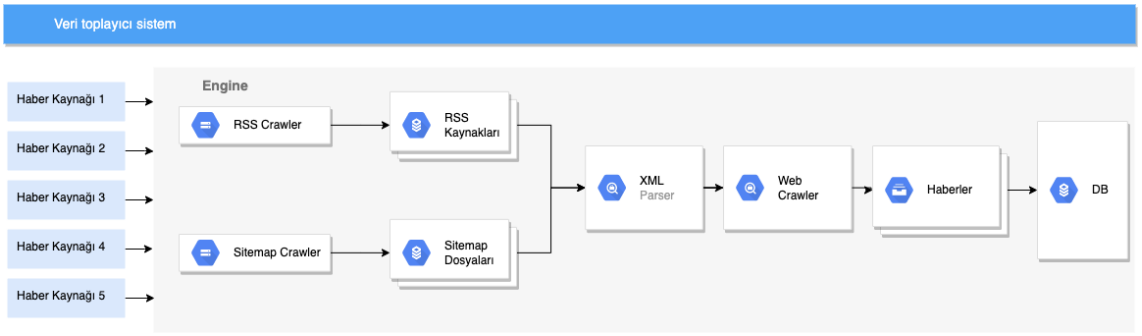
Gözetimsiz öğrenme algoritmaları genellikle verilerden elde ettiği istatistikleri kullanmaktadır. Kelime temsillerinde kullanılan CBOW ve Skip-gram modelleri gibi lokal bağlam penceresi yaklaşımları, külliyat içerisindeki kelimelerin anlamsal bilgileri hakkında bir sonuç çıkarabilmekte ancak kelimelerin tüm külliyat üzerinde birlikte bulunma sayıları yerine ayrı ayrı lokal pencereleri kullandığı için kelimelerin birlikte kullanılmalarına dair bir istatistik elde edememektedir. Gizli anlam analizi(LSA) gibi global matris çarpanlarına ayırma yaklaşımları ise külliyat üzerinden istatistiksel bilgileri etkili bir şekilde elde edebilmekte ancak kelimelerin anlamsal bilgilerinin belirlenmesi için yetersiz kalmaktadır. GloVe literatürdeki bu iki ana modelin avantajlarını birleştirerek kelimelerin vektör temsillerini oluşturmaktadır.

GloVe ile geniş bir külliyat kullanılarak eğitilen model vasıtasıyla, kelimeler arasında örneğin ülke-başkent, ülke-para birimi, lokasyon-bulunduğu ülke gibi ilişkiler çıkarılabilmektedir.

6. MALZEME VE YÖNTEM

6.1. Veri Toplama

Bu çalışmada kullanılmak üzere web kazıyıcı(scraper), web crawler gibi alt birimlerden oluşan bir veri toplama sistemi geliştirilmiş ve verilerin toplanması sağlanmıştır. Türkçe haber yayını yapan çok sayıda farklı kaynak bulunmaktadır. Bu kaynaklar içerisinden verilerin kolaylıkla toplanabilmesi için RSS yayını yapan haber kaynakları seçilmiştir. RSS genellikle haber siteleri, bloglar ve podcast yayınları tarafından kullanılan, bu kaynakların güncel içeriklerinin takip edilmesine olanak sağlayan bir besleme biçimidir. RSS yayınları RSS veya XML dosya biçimlerini kullanır. Yeni eklenen içerikler XML formatında ilgili kaynağın RSS bağlantısı üzerinden yayınlanır. Geliştirilen web kazıyıcı; CNN Türk, NTV, Habertürk, Sözcü ve Hürriyet internet sitelerinden oluşan 5 farklı haber kaynağının RSS yayınlarını kullanmaktadır. Sistem; Türkiye, Dünya, Siyaset, Gündem, Ekonomi, Magazin, Kültür-Sanat, Teknoloji, Spor ve Sağlık gibi farklı kategorilerindeki haberleri toplamış ve veritabanına kaydetmiştir. Veri toplayıcı sisteme dair akış Şekil 6.1.'de gösterilmektedir.



Şekil 6.1. Veri toplayıcı sisteme dair akış

Haber kaynaklarından haberler belirli zamanlarda yayınlanmadığı ve bir haberin herhangi bir zamanda yayınlanabileceği göz önünde bulundurulduğunda, anlık haber yayını takip etmek yerine saatlik periyotlarla yeni yayınlanan haberleri takip eden bir sistem geliştirilmiştir. Geliştirilen sistem her saat 5 haber kaynağının RSS yayınını kullanarak, sisteme yeni haberler kaydetmektedir. Geliştirilen web kazıyıcı, ilk olarak Kasım 2017 - Mayıs 2018 yılları arasında, sonrasında ise Aralık 2020 - Ocak 2022 tarihleri arasında çalıştırılarak veri toplamıştır. Ayrıca geliştirilen web kazıyıcının, veri setini genişletmek amacıyla, haber kaynaklarının site haritalarını kullanacak şekilde

modifiye edilmiş bir versiyonu çalıştırılarak farklı zamanlarda sistemin veri toplaması sağlanmıştır. RSS kaynağına gerçekleştirilen iki istek arasında, RSS kaynağında aynı haberin tekrarlı bir şekilde olabilmesi ihtimaline karşılık, geliştirilen sistem haberin kendisine özel URL kalıplarını hashleyerek aynı haberlerin tekrarlı bir biçimde kaydedilmesinin önüne geçecek şekilde geliştirilmiştir. Haber kaynaklarından elde edilen her bir haberin URL'i, haber kaydedilirken md5 algoritması ile hashlenmiş ve URL'in md5 özeti(hash) veritabanına kaydedilmiştir. Kaynaktan yayınlanan haberde bir güncelleme varsa güncelleme tarihi her kaynak için elde edilemediği için aynı URL'e sahip haberler tek bir haber olarak değerlendirilmiştir. Geliştirilen sistem ile, kaynaklardan elde edilen haber metinleri, haber başlığı, kaynak url'i ve yayınlandığı zaman bilgisi toplanmıştır. Farklı kaynaklardan toplanan haber metinleri sonrasında TextRank algoritması ile özetlenerek haberlerin ham metinlerine ek olarak, özetlenmiş haber metinleri de veri tabanına kaydedilmiştir.

Tablo 6.1. Toplanan veri istatistiği

Toplam Haber Sayısı	353.513
Ham Haber Metinlerindeki Toplam Kelime Sayısı	66.517.527
Özet Haber Metinlerindeki Toplam Kelime Sayısı	29.085.912
Ham Haber Başına Düşen Ortalama Kelime Sayısı	188,1
Özet Haber Başına Düşen Ortalama Kelime Sayısı	82,2

Tablo 6.2. Eğitim verisine dair veri istatistiği

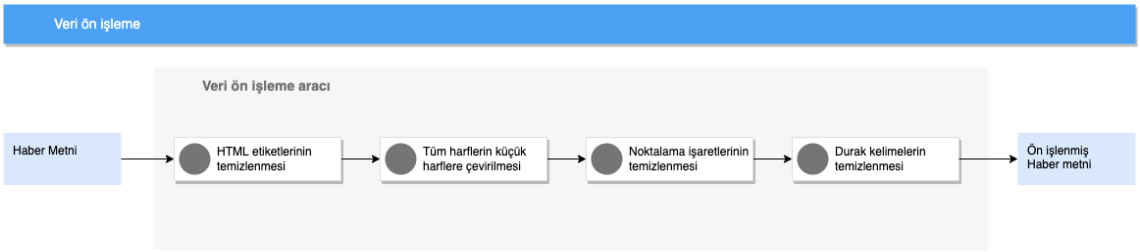
Toplam Haber Sayısı	28.302
Ham Haber Metinlerindeki Toplam Kelime Sayısı	5.581.840
Özet Haber Metinlerindeki Toplam Kelime Sayısı	2.506.477
Ham Haber Başına Düşen Ortalama Kelime Sayısı	197,2
Özet Haber Başına Düşen Ortalama Kelime Sayısı	88,5

Tablo 6.1.'de veri toplayıcı sistem tarafından toplanan veriye dair istatistikler gösterilmektedir. Veri toplayıcı sistem ile toplanan 350.000'in üzerindeki haber arasından, 2022 Ocak ayında 5 farklı kaynaktan yayınlanmış tüm haberler seçilerek eğitim verisi oluşturulmuştur. 2022 Ocak ayında CNN Türk, NTV, Habertürk, Sözcü ve Hürriyet haber sitelerinden sistem vasıtasıyla keşfedilen ve kaydedilen toplam haber

sayısı 28.302'dir. Eğitim verisi olarak seçilen veriye dair istatistik Tablo 6.2.'de gösterilmektedir.

6.2. Ön İşleme

Metin verileriyle çalışılırken, kaynaklardan elde edilen ham metinlerin bir ön işlemeye tabi tutulması gerekmektedir. Ham metinler içerisinde gereksiz terimler bulunabilir. Yazarlar genellikle yazıldığı dilin kurallarına tam olarak uymamaktadır. Kaynaklardan elde edilen ham metinlerde yarım bırakılmış sözcükler, yazım hataları, anlatım bozukluklarıyla karşılaşılabilmektedir. Geliştirilen ön işleme aracına dair akış Şekil 6.2.'de gösterilmektedir.



Şekil 6.2. Ön işleme aracına dair akış

Metin ön işleme sürecinde; HTML etiketlerin temizlenmesi, kelimelerin tüm harflerinin küçük harflere çevrilmesi, durak sözcüklerin temizlenmesi, noktalama işaretlerinin temizlenmesi ve noktalama işaretlerinin temizlenmesinden sonra boş kalabilecek karakterler yerine boşluk eklenmesi gibi bir takım işlemler gerçekleştirilmektedir. Ön işleme sürecinde, problemin türüne göre metnin cümlelere veya kelimelere bölünmesi işlemi de gerekli olabilmektedir. Bu sebeple geliştirilen ön işleme aracında haber metinlerini cümlelere ve haber metinlerini oluşturan cümleleri kelimelerine bölecek fonksiyonlar bulunmaktadır. Bir metin içerisinde tek başına bir anlam ifade etmeyen ve metin içerisinde sıkça tekrarlayan bağlaçlar, edatlar gibi kelimeler bulunur. Bu kelimelere durak kelimeler denmektedir. Durak kelimelerin metinden temizlenmesi o metnin ayırt ediciliğini bir kayba uğratmaz. Durak kelimeler tüm metinlerde yer alan kelimelerdir ve metinlerin sınıflandırılması veya özetlenmesi sürecinde sisteme olumlu bir katkı sunmazlar. Tablo 6.3.'de Türkçe durak kelimelerden örnekler gösterilmektedir. Geliştirilen sistem kullanılarak haber metinleri bir ön işleme sürecinden geçirilmiştir. Metinleri oluşturan tüm kelimeler küçük harflere çevrilmiştir. Metinlerde yer alan

noktalama işaretleri, durak kelimeler ve HTML etiketler temizlenerek veritabanına kaydedilmiştir.

Tablo 6.3. Türkçe durak kelimelerden örnekler

ama	bazen	gibi
ancak	beri	hala
artık	bile	hatta
aslında	biraz	herkes
ayrıca	böylece	işte

6.3. Otomatik Metin Özetleyici

Geliştirilen web kazıyıcı sistem ile kaydedilen haberler çıkarıma dayalı özetleme algoritmalarından olan TextRank algoritması ile özetlenmiştir. Veri toplayıcı sistem tarafından haber kaynaklarından elde edilen yalnızca görsel veya video içerik barındıran haberler, fotoğraf galerileri gibi çok sayıda sayfadan oluşan ve metin içermeyen haberler değerlendirmeye alınmamıştır. Tablo 6.4. ve Tablo 6.5.'de iki ayrı referans haber metninin ham haber metinleri ile birlikte TextRank algoritması ile özetlenmiş özet metinleri gösterilmektedir.

Tablo 6.4. Otomatik metin özetleyici ile özetlenen örnek metin - 1

Orijinal Metin	Rusya ile savaşın eşiğine gelen Ukrayna güne saldırı haberi ile uyandı. Ülkenin doğusundaki bir havacılık fabrikasına Silahlı Saldırı meydana geldi Ukrayna İçişleri Bakanlığı bir ulusal muhafızın fabrikada ateş açtığı olay nedeniyle 5 kinin öldüğünü duyurdu. Polis, Ukrayna'da bir ulusal muhafız askerinin Perşembe günü ülkenin doğusundaki bir füze ve roket fabrikasına ateş açarak 4 asker arkadaşını öldürdüğünü duyurdu. Olay nedeniyle bir sivilin hayatını kaybederken 5 kişi de yaraladı. SALDIRGAN KAYIPLARA KARIŞMIŞTI Dnipro şehrinde gerçekleşen olayda Kalaşnikof tüfeği ile saldırıyı düzenleyen kişinin saldırıdan sonra kayıplara karışmıştı. Artem Ryabchuk isimli askerin yakalanması için saatlerdir süren operasyon sonuç verdi. Bakanlık Ryabchuk'un yakalandığını duyurdu ve olay anından ilk görüntüyü basına servis etti. Güvenlik güçleri saldırganın Dnipro'nun dışındaki Pidgorodne kasabasında gözaltına alındığını söyledi. Saldırgan askerin Rusya ile bağlantılı olabileceği iddia edilirken bu iddia ile ilgili henüz resmi bir açıklama gelmedi. İçişleri Bakanı Denys Monastyrsky, 'Verdiğim emir doğrultusunda ülkesini savunmak ve güvenlikten sorumlu olmak üzere çağrılan 21 yaşındaki bir askerin bu eylemleri gerçekleştirmesine neden olan koşulları incelemek üzere bir komisyon kurulacak' dedi. EN ÖNEMLİ ASKERİ FABRİKALARDAN BİRİ Ülkenin en büyük şehirlerinden biri olan Dnipro'da bulunan fabrika, bir zamanlar kıtalararası balistik füzelerin üretim yeriydi.
Özet Metin	Ülkenin doğusundaki bir havacılık fabrikasına Silahlı Saldırı meydana geldi. Ukrayna İçişleri Bakanlığı bir ulusal muhafızın fabrikada ateş açtığı olay nedeniyle 5 kinin öldüğünü duyurdu. Polis, Ukrayna'da bir ulusal muhafız askerinin Perşembe günü ülkenin doğusundaki bir füze ve roket fabrikasına ateş açarak 4 asker arkadaşını öldürdüğünü duyurdu. Olay nedeniyle bir sivilin hayatını kaybederken 5 kişi de yaraladı. Bakanlık Ryabchuk'un yakalandığını duyurdu ve olay anından ilk görüntüyü basına servis etti.

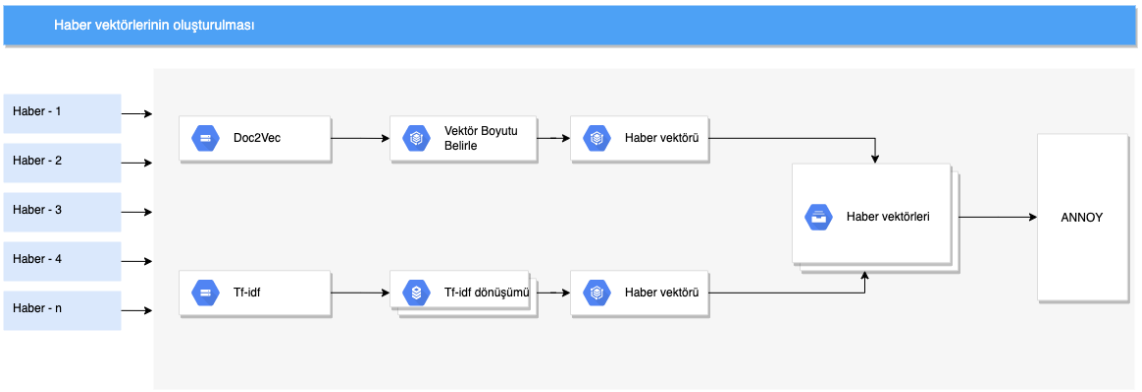
Tablo 6.5. Otomatik metin özetleyici ile özetlenen örnek metin - 2

Orijinal Metin	İtalya Sağlık Bakanı Roberto Speranza yaptığı açıklamada, Avrupa Birliği (AB) ülkelerinden gelen tüm ziyaretçilere uygulanan Covid-19 kısıtlamalarının 1 Şubat'tan itibaren hafifletileceğini duyurdu. Bu kapsamda Bakan Speranza, AB ülkelerinden gelen yolcuların sadece "yeşil geçiş" belgesiyle ülkeye girebilmesini mümkün kılan düzenlemeyi imzaladı. Speranza'nın imzaladığı düzenlemeyle aynı zamanda AB dışındaki 6 ülkeye de turistik amaçlı seyahat izni verildi. Türkiye, Küba, Singapur, Tayland (Phuket adasıyla sınırlı), Umman ve Fransız Polinezyası "Covidsiz turizm koridoruna" dahil edilirken, bu ülkelere yapılacak turizm amaçlı seyahatlere şartlar dahilinde müsaade edildi. KARANTİNA GEREKMİYOR Türkiye'nin de aralarında bulunduğu 6 ülkeye turistik seyahat için yolcuların seyahatlerden en fazla 48 saat önce yapılmış test sonucu, İtalya'ya girişte havalimanında da test yaptırması ve negatif sonuç ibraz etmesi gerekiyor. Söz konusu kuralları yerine getirenlerin ayrıca Covid-19 aşısı yaptırdıklarını ya da hastalığı geçirip iyileştiklerini belgelemeleri gerekiyor. İtalya'ya dönüşte ise karantinaya girmeleri gerekmiyor. İtalya, bugüne kadar Türkiye'ye yalnızca sağlık, iş, ikamet adresine dönüş gibi nedenlerle seyahate izin veriyordu. Aşıya bakılmaksızın tüm yolcuların İtalya'ya dönüşte 10 gün karantinada kalması gerekiyordu. İtalya Sağlık Bakanlığına göre ülkede çarşamba günü 167 bin 206 yeni korona virüs vakası tespit edilirken, 426 kişi hayatını kaybetti.
Özet Metin	İtalya Sağlık Bakanı Roberto Speranza yaptığı açıklamada, Avrupa Birliği (AB) ülkelerinden gelen tüm ziyaretçilere uygulanan Covid-19 kısıtlamalarının 1 Şubat'tan itibaren hafifletileceğini duyurdu. Bu kapsamda Bakan Speranza, AB ülkelerinden gelen yolcuların sadece "yeşil geçiş" belgesiyle ülkeye girebilmesini mümkün kılan düzenlemeyi imzaladı. Speranza'nın imzaladığı düzenlemeyle aynı zamanda AB dışındaki 6 ülkeye de turistik amaçlı seyahat izni verildi. Türkiye'nin de aralarında bulunduğu 6 ülkeye turistik seyahat için yolcuların seyahatlerden en fazla 48 saat önce yapılmış test sonucu, İtalya'ya girişte havalimanında da test yaptırması ve negatif sonuç ibraz etmesi gerekiyor. Aşıya bakılmaksızın tüm yolcuların İtalya'ya dönüşte 10 gün karantinada kalması gerekiyordu.

TextRank algoritması ile haber metinlerinin özetlenmesi ile birlikte özet haber metinlerinin kullanılarak benzerlik tespitinde kullanılacak haber vektör boyutlarının düşürülerek işlem maliyetinin düşürülmesi ve benzerlik tespiti yapılırken başarının artırılması hedeflenmiştir.

6.4. Haber Vektörlerinin Oluşturulması

Toplanan haberler üzerinde benzerlik tespiti yapılması için haber metinlerinin vektör uzayında temsil edilmesi gerekmektedir. Metin vektörleştirmesi metinlerin sayısal olarak temsil edilmesini sağlamak adına yapılan işlemleri ifade etmektedir.



Şekil 6.3. Haber vektörlerinin oluşturulmasına dair akış

Haber vektörleştirmesi yapılırken 2022 Ocak ayında 5 farklı kaynaktan yayınlanan toplamda 28.000 haberin ham metinleri ve özetlenen haber metinleri kullanılarak, ayrı ayrı TF-IDF ve Doc2Vec ile haber vektörleri oluşturulmuştur. Bu sayede her bir haberin tüm veri kümesi üzerinde birer vektör olarak temsil edilmesi sağlanmıştır. Haber vektörlerinin oluşturulmasına dair akış Şekil 6.3.'de gösterilmektedir.

Haber vektörleri oluşturulurken; TF-IDF dönüşümleri için scikit-learn kütüphanesi, Doc2Vec vektörlerinin elde edilmesi için Gensim Doc2Vec kütüphanesi kullanılmıştır. TF-IDF ile haber metinleri vektörleştirilirken, TF-IDF dönüşümleri hesaplanmış ve her bir haber vektörü ayrı ayrı elde edilmiştir.

Tablo 6.7. Örnek TF-IDF matrisi

	kelime-1	**kelime-2**	**kelime-3**	**…**	**kelime-n**
Haber-1	t1-1	t1-2	t1-3	..	t1-n
Haber-2	t2-1	t2-2	t2-3	..	t2-n
…	..	..	..	..	..
Haber-n	tn-1	tn-2	tn-3	..	tn-n

Haber vektörleri oluşturulurken TF-IDF vektörleri ham haber metinleri ve özet haber metinleri kullanılarak iki ayrı şekilde vektörleştirilmiştir. Tablo 6.7.'de örnek bir TF-IDF matrisi gösterilmektedir.

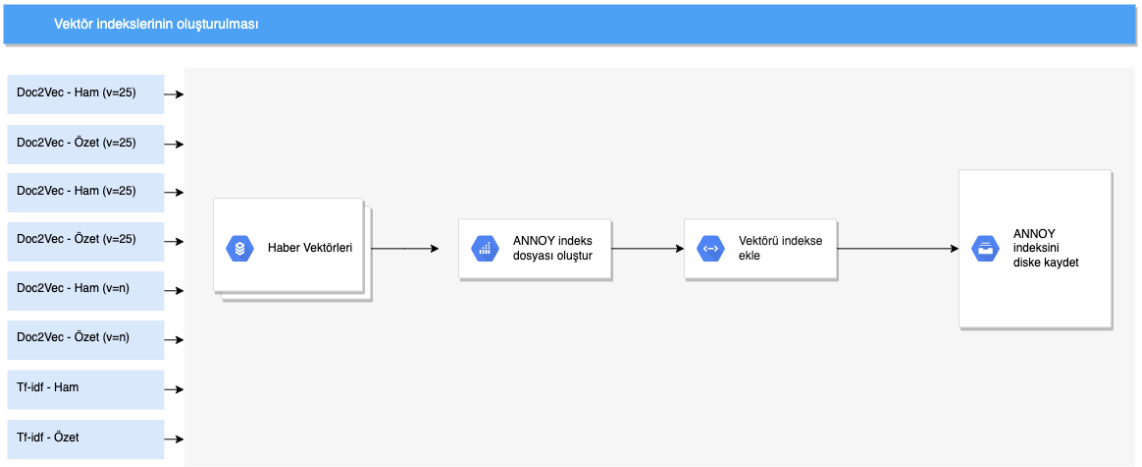
Tablo 6.8. Örnek Doc2Vec vektör matrisi

	1	**2**	**3**	**…**	**v= vektör boyutu**
Haber-1	Vektör[1][0]	Vektör[1][1]	Vektör[1][2]	..	Vektör[1][v-1]
Haber-2	Vektör[2][0]	Vektör[2][1]	Vektör[2][2]	..	Vektör[2][v-1]
…	..	..	..	..	..
Haber-n	Vektör[n][0]	Vektör[n][1]	Vektör[n][2]	..	Vektör[n][v-1]

Doc2Vec ile vektörleştirme yapılırken ise ayrı ayrı; vektör boyutu 25, 40, 50, 60, 75 ve 100 olacak şekilde hem ham haber metinleri kullanılarak 6 farklı şekilde, hem de özet haber metinleri kullanılarak 6 farklı şekilde vektörleştirme yapılmıştır. Doc2Vec ile toplamda 12 farklı şekilde vektörleştirme yapılmıştır. Tablo 6.8.'de Doc2Vec vektörlerinin örnek bir matris üzerinde temsili gösterilmektedir.

6.5. Vektör İndekslerinin Oluşturulması

Haber vektörleri oluşturulduktan sonra bir habere benzer olan haberlerin tespit edilmesi için yaklaşık en yakın komşu araması yapabilmek amacıyla vektörlerin indeks dosyaları halinde saklanması gerekmektedir. Annoy kütüphanesi ile haber vektörleri kullanılarak en yakın komşu araması yapılması için, her bir vektörleştirme yönteminde ayrı ayrı indeks dosyaları oluşturulması gerekmektedir. Bu sebeple oluşturulan haber vektörleri ile, Annoy kütüphanesi kullanılarak TF-IDF ve Doc2Vec için oluşturulan her bir vektör modeli için ayrı ayrı olacak şekilde Annoy indeksleri oluşturulmuş ve oluşturulan Annoy indeksleri dosya sistemine kaydedilmiştir. Vektör indekslerinin oluşturulmasına dair akış Şekil 6.4.'de gösterilmektedir.



Şekil 6.4. Vektör indekslerinin oluşturulmasına dair akış

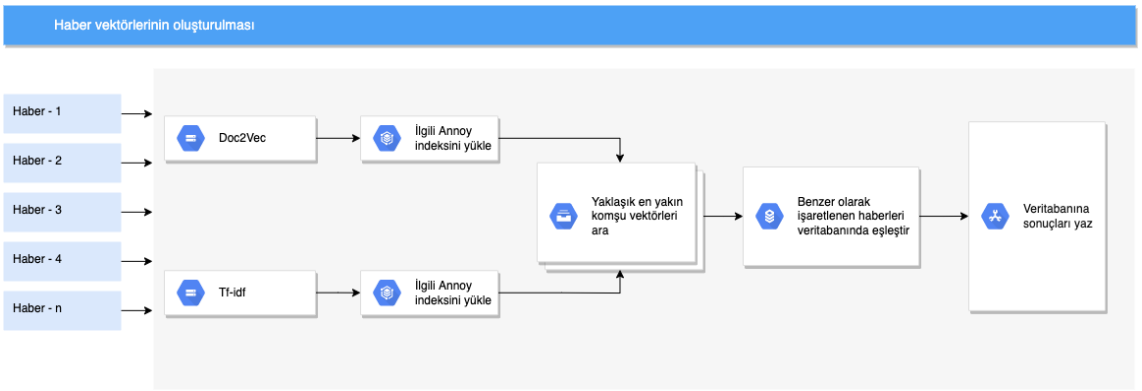
Bir habere en benzer haberler tespit edilirken sistem mesafe hesaplamalarını indeks üzerinden yaptığı için, indeks oluşturulurken kullanılan sıralama önemlidir. Veri setindeki haberler sıralı bir şekilde indekslere eklenmiş ve bu sayede veritabanı ile aynı sıralamayı kullanması sağlanmıştır. Bu sayede Annoy indeksi üzerinden, herhangi bir haberin sıra numarası kullanılarak o habere ait hesaplanan vektör elde edilebilmekte ve habere ait vektör kullanılarak diğer haber vektörleri ile arasındaki mesafe kosinüs benzerliği ile hesaplanabilmektedir.

Geliştirilen sistem ile 2022 Ocak ayında yayınlanmış 28.000 haber metni ile, ayrı ayrı ham haberler ve özet haberler kullanılarak TF-IDF dönüşümü ile TF-IDF matrisleri elde edilmiştir. Elde edilen matrisler kullanılarak, her bir haber vektörü Annoy kütüphanesi

ile indeks üzerine eklenmiş ve TF-IDF Annoy indeksleri oluşturularak dosya sistemine kaydedilmiştir. Annoy indeksleri oluşturulurken TF-IDF vektörleri ham haber metinleri ve özet haber metinleri kullanılarak iki ayrı şekilde indeks oluşturulmuştur. Doc2Vec ile ise ayrı ayrı; vektör boyutu 25, 40, 50, 60, 75 ve 100 olacak şekilde hem ham haber metinleri hem de özet haber metinleri kullanılarak toplamda 12 farklı indeks oluşturulmuştur.

6.6. Benzerlik Tespiti

Bu çalışmada, veri toplayıcı sistem tarafından toplanan haberler arasında benzer haberlerin tespit edilmesi için bir sistem geliştirilmiştir. Sistem Python dili ile geliştirilmiştir. Geliştirilen sistem; TF-IDF dönüşümleri için scikit-learn kütüphanesini, Doc2Vec vektörlerinin elde edilmesi için Gensim Doc2Vec kütüphanesini, MariaDB veritabanı ile bağlantı için MySQL kütüphanesini, benzerlik tespiti için ise Annoy kütüphanesini kullanmaktadır. Haber vektörleri arasındaki benzerliklerin hesaplanması için, Annoy indeksleri üzerinde Kosinüs benzerlikleri kullanılarak en yakın komşu araması yapılmıştır. Rastgele seçilen 15 haber için, 2022 Ocak ayında yayınlanan 28.000 haber kullanılarak ayrı ayrı oluşturulan 14 farklı Annoy indeksi üzerinde en yakın komşu araması yapılmış, her bir vektörleştirme yöntemi için, seçilen haberlere en yakın mesafedeki haberler tespit edilmiş ve kullanılan yöntem parametreleri ile birlikte veritabanı üzerine kaydedilmiştir. Benzer haberlerin tespit edilmesi sürecine dair akış Şekil 6.5.'de gösterilmektedir.



Şekil 6.5. Benzer haberlerin tespit edilmesi sürecine dair akış

Benzerlik tespiti için geliştirilen sistemde, oluşturulan Annoy indeksi ve vektörler arasındaki mesafeler, kosinüs benzerliği ile ölçülerek haberler arasındaki benzerlik

hesaplanmaktadır. Geliştirilen sistem veri seti üzerindeki herhangi bir habere, en benzer n adet haberi tespit edebilmektedir. Tablo 6.9. ve 6.10.'da referans olarak seçilen iki ayrı haber için yaklaşık en yakın komşuları hesaplanarak elde edilen benzer haberlere ait haber başlıkları gösterilmektedir.

Tablo 6.9. Tespit edilen benzer haber örnekleri-1

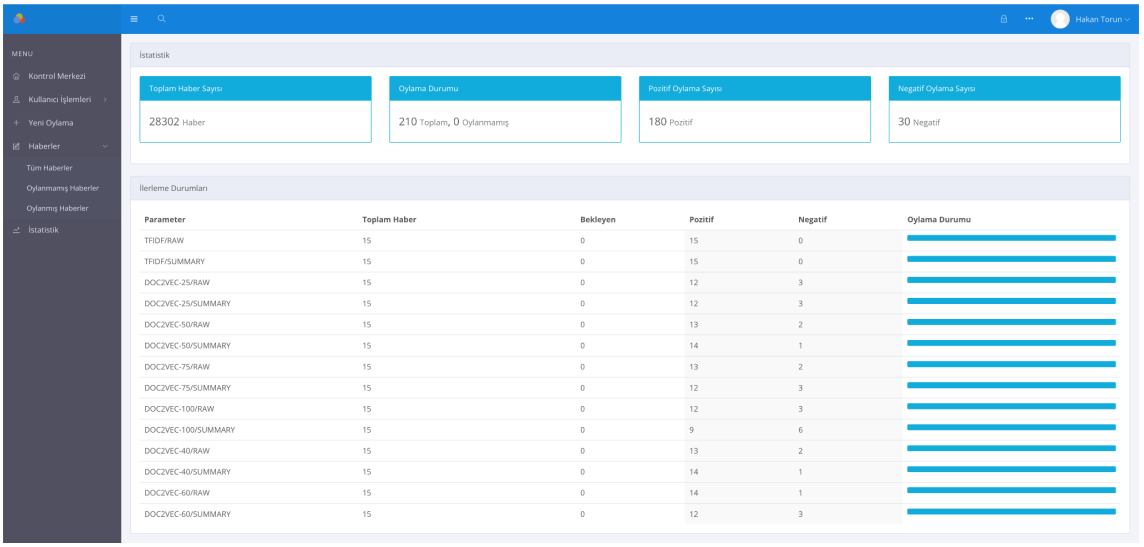
	Haber Başlığı
Referans	İtalya, Türkiye'ye seyahat kısıtlamasını gevşetiyor
Haber-1	Orta Avrupa'da omicron dalgası
Haber-2	İtalya, Türkiye ile seyahat kısıtlamalarını gevşetiyor: Turistik gezilere izin verilecek
Haber-3	İtalya'dan flaş Türkiye kararı! Şubattan itibaren geçerli olacak
Haber-4	Akdeniz'de göçmen teknesinde can pazarı

Tablo 6.10. Tespit edilen benzer haber örnekleri-2

	Haber Başlığı
Referans	Biden'dan Putin'e: Ukrayna'yı işgal halinde bedel ödersiniz
Haber-1	ABD Başkanı Biden ve Ukrayna Başkanı Zelenskiy görüştü
Haber-2	Putin ve Biden'ın görüşmesi öncesi 'Ukrayna ile dayanışma' mesajı
Haber-3	'Saldırı hazırlığı' iddiaları! Sıcak bölgeyle ilgili Ukrayna'dan açıklama
Haber-4	ABD, Ukrayna sınırındaki Rus askeri hareketliliğinden kaygılı

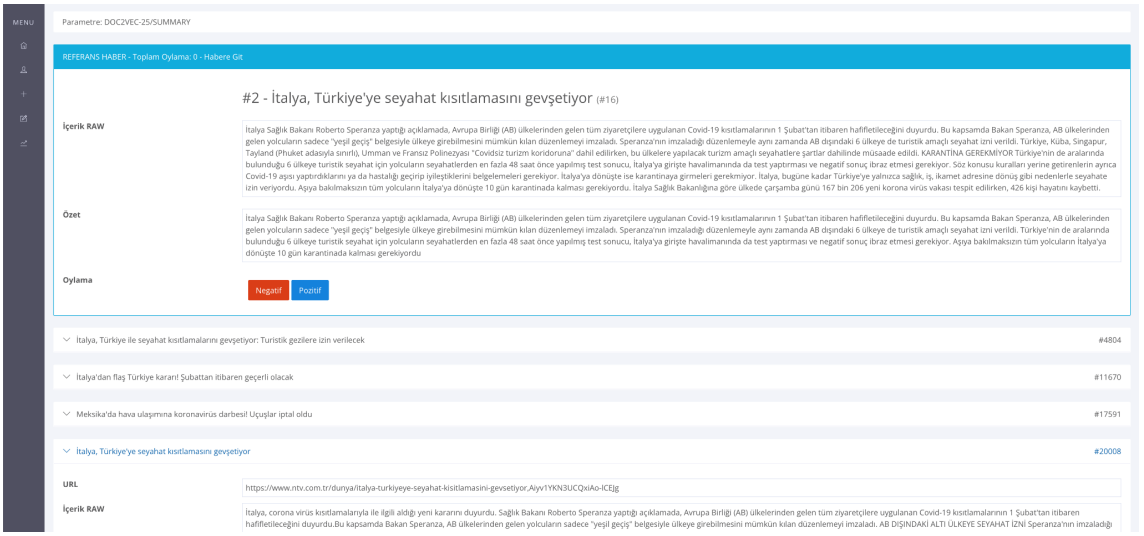
6.7. Performans Analizi

Sistemin başarısının ölçülmesi için, uzmanlar tarafından değerlendirme yapılması amacıyla Şekil 6.6.'da gösterilen web tabanlı bir oylama sistemi geliştirilmiş ve benzer olarak tespit edilen haberler uzmanların oylamasına sunulmuştur. Uzmanlar bir habere benzer olarak tespit edilen haberleri, oylama ekranını kullanarak referans habere ait tüm detaylar ile birlikte görüntüleyip oylama işlemi gerçekleştirmektedir. Geliştirilen web tabanlı sistemde her bir uzman için ayrı ayrı oylama yapılmasına imkan sağlayacak şekilde bir altyapı tasarlanmıştır. Sisteme eklenen uzmanlar kendi kullanıcı bilgileri ile sisteme giriş yaparak oylama yapabilmektedir. Bu sayede birden fazla sayıda uzman tarafından oylama yapılabilmekte ve elde edilen sonuçların her bir uzman için ayrı ayrı değerlendirmesi yapılabilmektedir.



Şekil 6.6. Uzman oylaması için geliştirilen web tabanlı sistem - 1

Geliştirilen oylayıcı sistem ile farklı parametrelerle birlikte oylanan haberler sisteme kaydedilmektedir. Bu sayede örneğin yeni bir kelime gömme modeliyle haber vektörleri elde edildikten sonra, benzer haberler tespit edilerek oylayıcı sisteme kaydedilmekte ve uzmanlar tarafından kolaylıkla oylama yapılabilmektedir.



Şekil 6.7. Uzman oylaması için geliştirilen web tabanlı sistem - 2

Şekil 6.7.'de uzman oylaması için geliştirilen sistemde, Doc2Vec ile, özet haberler kullanılarak vektör boyutunun 25 olması durumunda, vektörleştirilen haber metinleri arasında yaklaşık en yakın komşu araması ile tespit edilen benzer haberlerin uzman oylamasına sunulduğu oylama ekranı gösterilmektedir. Uzmanlar, referans habere benzer olarak tespit edilen haberler içerisinden en az bir adet benzer haber bulunması

durumunda işlemi başarılı olarak değerlendirmiş, tespit edilen haberler içerisinde referans habere benzer bir haber bulunmaması durumunda ise başarısız olarak değerlendirme yapmıştır. Uzmanların oylaması sonucunda elde edilen sonuçlar kullanılarak sistemin başarısının ölçülmesi, TF-IDF ve Doc2Vec yöntemlerinin karşılaştırılması, Doc2Vec vektör boyutlarının başarıya etkisi ile birlikte özet haberlerin ve ham haberlerin benzerlik tespiti için kullanılması durumunda sistem performanslarının karşılaştırılması hedeflenmiştir.

7. BULGULAR VE TARTIŞMA

Bu tez çalışmasında haber metinleri arasındaki benzerliklerin tespit edilmesi için, haber metinlerinin vektör uzayında temsil edilerek haber vektörleri arasındaki mesafelerin ölçülmesi ile Annoy kütüphanesi kullanılarak en yakın komşu aramasını kullanan bir yöntem geliştirilmiştir. Ham haber metinleri ve özet haber metinleri kullanılarak, ayrı ayrı TF-IDF ve Doc2Vec vektörleştirme yöntemleri ile oluşturulan haber vektörleri üzerinde en yakın komşular Kosinüs benzerliği kullanılarak tespit edilmiş ve karşılaştırmalı olarak performansları incelenmiştir.

Benzerlik tespiti için geliştirilen sistemde, Kosinüs benzerliği kullanılarak; Annoy indekslerinin oluşturulma süresi, oluşturulan Annoy indeksi üzerinde arama yapılarak veri setinde bulunan bir habere en benzer 5 adet haberin bulunması için geçen süreler ve indeks boyutları karşılaştırmalı olarak Tablo 7.1.'de yer almaktadır. Tabloda yer alan TF-IDF matris boyutu veri setindeki haber metinlerinin TF-IDF dönüşümlerinin hesaplanması ile elde edilen matrisin boyutunu, Doc2Vec matris boyutu veri setindeki haber metinlerinden elde edilen Doc2Vec vektörlerinden oluşan matris boyutunu, Annoy indeks boyutu oluşturulan Annoy indekslerinin disk üzerinde kaç GB yer kapladığını, indeks oluşturma süresi saniye cinsinden Annoy indekslerinin oluşturulma süresini, arama süresi ise oluşturulan Annoy indeksi üzerinde bir haber vektörüne en yakın 5 haber vektörünün aranması için geçen süreyi ifade etmektedir.

Tablo 7.1. Benzerlik tespiti mesafe ölçütlerine göre performans karşılaştırması

HAM HABER METİNLERİ / TF-IDF	
TF-IDF Matris Boyutu	248.886 x 25.000
Annoy indeks boyutu	24,91 GB
İndeks Oluşturma Süresi	220 Saniye
Arama Süresi	80 Saniye
ÖZET HABER METİNLERİ / TF-IDF	
TF-IDF Matris Boyutu	128.354 x 25.000
Annoy indeks boyutu	12,85 GB
İndeks Oluşturma Süresi	106 Saniye
Arama Süresi	38 Saniye

Tablo 7.1. (Devam) Benzerlik tespiti mesafe ölçütlerine göre performans karşılaştırması

HAM HABER METİNLERİ / DOC2VEC / Vektör Boyutu = 25	
Doc2Vec Matris Boyutu	25 x 25.000
Annoy indeks boyutu	386 MB
İndeks Oluşturma Süresi	108 Saniye
Arama Süresi	0.9 Saniye
ÖZET HABER METİNLERİ / DOC2VEC / Vektör Boyutu = 25	
Doc2Vec Matris Boyutu	25 x 25.000
Annoy indeks boyutu	381,2 MB
İndeks Oluşturma Süresi	47 Saniye
Arama Süresi	0.9 Saniye

HAM HABER METİNLERİ / DOC2VEC / Vektör Boyutu = 40	
Doc2Vec Matris Boyutu	40 x 25.000
Annoy indeks boyutu	412,4 MB
İndeks Oluşturma Süresi	93 Saniye
Arama Süresi	0.9 Saniye
ÖZET HABER METİNLERİ / DOC2VEC / Vektör Boyutu = 40	
Doc2Vec Matris Boyutu	40 x 25.000
Annoy indeks boyutu	408,5 MB
İndeks Oluşturma Süresi	51 Saniye
Arama Süresi	0.9 Saniye

HAM HABER METİNLERİ / DOC2VEC / Vektör Boyutu = 50	
Doc2Vec Matris Boyutu	50 x 25.000
Annoy indeks boyutu	429,7 MB
İndeks Oluşturma Süresi	108 Saniye
Arama Süresi	1 Saniye
ÖZET HABER METİNLERİ / DOC2VEC / Vektör Boyutu = 50	
Doc2Vec Matris Boyutu	50 x 25.000
Annoy indeks boyutu	425,8 MB
İndeks Oluşturma Süresi	54 Saniye
Arama Süresi	1 Saniye

Tablo 7.1. (Devam) Benzerlik tespiti mesafe ölçütlerine göre performans karşılaştırması

HAM HABER METİNLERİ / DOC2VEC / Vektör Boyutu = 60	
Doc2Vec Matris Boyutu	60 x 25.000
Annoy indeks boyutu	442,1 MB
İndeks Oluşturma Süresi	105 Saniye
Arama Süresi	1 Saniye
ÖZET HABER METİNLERİ / DOC2VEC / Vektör Boyutu = 60	
Doc2Vec Matris Boyutu	60 x 25.000
Annoy indeks boyutu	444,4 MB
İndeks Oluşturma Süresi	61 Saniye
Arama Süresi	1 Saniye

HAM HABER METİNLERİ / DOC2VEC / Vektör Boyutu = 75	
Doc2Vec Matris Boyutu	75 x 25.000
Annoy indeks boyutu	456,5 MB
İndeks Oluşturma Süresi	114 Saniye
Arama Süresi	1.4 Saniye
ÖZET HABER METİNLERİ / DOC2VEC / Vektör Boyutu = 75	
Doc2Vec Matris Boyutu	75 x 25.000
Annoy indeks boyutu	463,2 MB
İndeks Oluşturma Süresi	105 Saniye
Arama Süresi	1.3 Saniye

HAM HABER METİNLERİ / DOC2VEC / Vektör Boyutu = 100	
Doc2Vec Matris Boyutu	100 x 25.000
Annoy indeks boyutu	465,5 MB
İndeks Oluşturma Süresi	130 Saniye
Arama Süresi	1.8 Saniye
ÖZET HABER METİNLERİ / DOC2VEC / Vektör Boyutu = 100	
Doc2Vec Matris Boyutu	100 x 25.000
Annoy indeks boyutu	470,5 MB
İndeks Oluşturma Süresi	75 Saniye
Arama Süresi	1.7 Saniye

Uzman değerlendirmesine sunulan referans haberlere benzer olarak tespit edilen haberlerin oylanması sonucunda elde edilen sonuçlar sonucunda vektörleştirme yöntemlerine göre sistem başarısı Tablo 7.2.'de gösterilmektedir. Yöntem sütununda TF-IDF ve Doc2Vec vektörleştirme yöntemlerinden hangisinin kullanıldığı, bu yöntemle birlikte özet haber mi ham haber mi kullanıldığı bilgisi ve Doc2Vec vektör boyutları yer almaktadır. Örnek olarak; "TF-IDF / Ham Haber" haber metinlerinin TF-IDF ile vektörleştirildiği ve ham haber metinlerinin kullanıldığını, "Doc2Vec - 50 / Özet Haber" ise haber metinlerinin Doc2Vec ile vektörleştirildiğini, özet haberlerin kullanıldığını ve vektör uzunluğunun 50 olduğunu ifade etmektedir.

Tablo 7.2. Vektörleştirme yöntemlerine göre sistem başarısı

Yöntem	Toplam	Doğru Tespit Edilme Sayısı	Doğru Tespit Edilmeme Sayısı
TF-IDF / Ham Haber	15	**15**	0
TF-IDF / Özet Haber	15	**15**	0
Doc2Vec - 25 / Ham Haber	15	12	3
Doc2Vec - 25 / Özet Haber	15	12	3
Doc2Vec - 40 / Ham Haber	15	13	2
Doc2Vec - 40 / Özet Haber	15	**14**	1
Doc2Vec - 50 / Ham Haber	15	13	2
Doc2Vec - 50 / Özet Haber	15	**14**	1
Doc2Vec - 60 / Ham Haber	15	**14**	1
Doc2Vec - 60 / Özet Haber	15	12	3
Doc2Vec - 75 / Ham Haber	15	13	2
Doc2Vec - 75 / Özet Haber	15	12	3
Doc2Vec - 100 / Ham Haber	15	12	3
Doc2Vec - 100 / Özet Haber	15	9	6

Benzer haberlerin tespit edilmesi için geliştirilen sistemde özet haber metinlerinin kullanılması durumunda vektör indekslerinin oluşturulma süresi, indeks boyutu ve bu vektör uzayında yapılan arama süresinde önemli ölçüde hız kazanıldığı tespit edilmiştir.

Çalışmada yapılan uzman oylaması sonucunda elde edilen sonuçlar değerlendirildiğinde ise; haberler arasındaki benzerlik tespiti için TF-IDF vektörlerinin kullanılması

durumunda özet haber vektörlerinde indeks boyutunun düşmesi ile birlikte arama hız performansının önemli derecede arttığı tespit edilmiştir. Doc2Vec vektörlerinin kullanılması durumunda ise en iyi başarının vektör boyutunun 40 ile 60 arasında olması halinde gerçekleştiği sonucuna varılmıştır.

Haberler arasındaki benzerliklerin tespit edilmesinde TF-IDF vektörlerinin kullanılması durumunda, vektörlerin yüksek boyutlu matrisler şeklinde ifade edilmesi ve vektör indekslerinin yüksek boyutlu olması sebebiyle benzer haberlerin tespit edilmesi yüksek işlem gücüne ihtiyaç duymaktadır. Bu çalışmada; ham haber metinleri kullanılarak TF-IDF ile elde edilen haber vektörleri 257.435 x 28.000 boyutlu bir matris üzerinde temsil edilip Annoy indeks boyutu 29,16 GB olurken, aynı şekilde ham haber metinleri Doc2Vec vektörleri ile vektör boyutunun en yüksek seçildiği 100 boyutlu vektörlerde 100 x 28.000 boyutlu bir matris üzerinde temsil edilebilmekte ve Annoy indeks boyutu 466,5 MB olmaktadır. Bu sebeple, Doc2Vec vektörleri sabit uzunluklu vektörler ve TF-IDF'e göre çok küçük boyutlu vektörler olduğu için indeks üzerinde arama zamanı ve vektör indeks boyutları açısından avantajlıdır. Benzer haberlerin otomatik olarak tespit edilmesi ayrıca, medya takip ajansları gibi kuruluşların karşılaştığı önemli bir problemdir ve gerçek zamanlı benzer haberleri tespit edecek bir sistemin geliştirilmesi durumunda, en yakın komşuların aranması sürecindeki hız performansı önemlidir. Bu sebeple gerçek zamanlı sistemlerde, haberlerin benzerlik tespitinde özetleme ihtiyacının gerekli olduğu düşünülmektedir.

8. SONUÇLAR VE ÖNERİLER

Bilgi işlem kapasitesinin artması ile birlikte gerek makine öğrenmesi, gerek doğal dil işleme görevlerine dair problem çözümleri, araştırmacılar tarafından her geçen gün daha başarılı sonuçlar üretecek şekilde geliştirilmektedir. Makine öğrenmesi ve doğal dil işleme yöntemleri kullanılarak metin özetleme ve sınıflandırma gibi problemler için çözümler üretilmektedir. Bilgisayar bilimlerindeki bir diğer önemli problem ise yaklaşık en yakın komşuların aranması problemidir.

Bu çalışmada Türkçe yayın yapan haber kaynaklarından 350.000'den fazla haber metni geliştirilen web kazıyıcı ile toplanmış ve özetlenmiş, toplanan veriler arasından 2022 yılı Ocak ayı boyunca yayınlanan toplam 28.000 haber kullanılarak haber metinleri arasında benzerlik tespiti yapılması amacıyla bir sistem geliştirilmiştir. Haber metinlerinin özetlenmesi için çıkarıma dayalı özetleme yöntemlerinden Textrank algoritmasını kullanan bir otomatik metin özetleyici sistem geliştirilmiş ve haber metinlerinin özetlenmesi sağlanmıştır. TextRank algoritmasını kullanan özetleyici sistem, haber metinlerinin özetlenmesi için başarılı bir şekilde kullanılmış olmakla birlikte geliştirilen özetleyici sistemin haber metinleri haricindeki Türkçe metinlerin özetlenmesinde de kullanılabileceği düşünülmektedir.

Haber metinlerinin vektör uzayında temsil edilmesi ve haber vektörleri arasında kosinüs benzerliği kullanılarak en yakın komşu araması yapılarak kaynaklardan yayınlanan benzer haberlerin tespit edilebileceği bir sistemin geliştirilmesinin başarılı bir şekilde kullanılabileceği sonucuna varılmıştır. Elde edilen sonuçlar değerlendirildiğinde TF-IDF vektörlerinin farklı kaynaklardan yayınlanan benzer haberleri tespit etmek için kullanılabileceği, Doc2Vec vektörlerinin ise hem farklı kaynaklardan yayınlanan benzer haberler hem de semantik olarak ilişkili haberlerin tespitinde kullanılabileceği düşünülmektedir.

Çalışmalar gerçekleştirilirken yapılan gözlemlerde, haberler arasındaki benzerliğin aynı zamanda birbiriyle ilişkili haberlerin tespitinde de kullanılabileceği düşünülmektedir. Benzer haberlerin yayınlandıkları zaman bilgileri ve haberlerin kategorileri kullanılarak gruplandığında benzer haberler ve birbiriyle ilişkili haberler olarak tespit edilebileceği düşünülmektedir.

Türkçe haber metinleri kullanılarak yapılan çalışmalarda kullanılan veri setleri genellikle güncel olmayan veriler içermektedir. Araştırmacılar kimi zaman kısıtlı sayıda veri içeren veri setlerini kullanmak zorunda kalmaktadır. 2018 yılında yaptığımız çalışmada 12.000 Türkçe haber metnini geliştirdiğimiz bir web kazıyıcı ile farklı haber kaynaklarından toplayarak çalışmıştık (Torun ve İnner, 2018). Bu tez çalışmasında ise 350.000'in üzerinde Türkçe haber metni geliştirdiğimiz web kazıyıcı ve veri toplama aracı ile toplanarak kullanılmıştır. Daha önceki çalışmamızda, farklı kaynaklardan yayınlanan haberler arasında benzerlik tespiti yapmış ve yalnızca farklı kaynaklardan yayınlanan benzer haber çiftlerini tespit edebilen bir sistem geliştirmiştik. Bu tez çalışmasında ise, farklı kaynaklardan yayınlanan yüksek sayıdaki haber metinleri içerisinden bir habere en yakın n adet benzer haberin tespit edilmesini sağlayan bir sistem geliştirilmiştir.

KAYNAKLAR

Altan, Z. (2004). A Turkish automatic text summarization system. In *IASTED International Conference on AIA*, 16-18.

Andoni, A., Indyk, P., Laarhoven, T., Razenshteyn, I., & Schmidt, L. (2015). Practical and optimal LSH for angular distance. *Advances in neural information processing systems,* 28.

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, *5*, 135-146.

Brin, S., & Page, L. (1998). The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems*, *30*(1-7), 107-117.

Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. *IEEE transactions on information theory*, *13*(1), 21-27.

Çayıroğlu, İ. (2015). Yapay Sinir Ağları. *Karabük Üniversitesi, Karabük, Türkiye*.

Dumais, S. T. (2004). Latent semantic analysis. *Annual review of information science and technology*, *38*(1), 188-230.

Edmundson, H. P., & Wyllys, R. E. (1961). Automatic abstracting and indexing—survey and recommendations. *Communications of the ACM*, *4*(5), 226-234.

Edmundson, H. P. (1969). New methods in automatic extracting. *Journal of the ACM (JACM)*, *16*(2), 264-285.

Eduard, H., & Lin, C. Y. (1998). Automated text summarization and the SUMMARIST system. In *Proceedings of a workshop*, 197-214.

Erkan, G., & Radev, D. R. (2004). Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research*, *22*, 457-479.

Fattah, M. A., & Ren, F. (2008). Automatic text summarization. *World Academy of Science, Engineering and Technology, 37*(2), 192.

Genest, P. E., & Lapalme, G. (2011). Framework for abstractive summarization using text-to-text generation. In *Proceedings of the workshop on monolingual text-to-text generation*, 64-73.

Gupta, V., & Lehal, G. S. (2010). A survey of text summarization extractive techniques. *Journal of emerging technologies in web intelligence*, *2*(3), 258-268.

Houle, M. E., & Nett, M. (2014). Rank-based similarity search: Reducing the dimensional dependence. *IEEE transactions on pattern analysis and machine intelligence*, *37*(1), 136-150.

Kaelbling, L. P., Littman, M. L., & Moore, A. W. (1996). Reinforcement learning: A survey. *Journal of artificial intelligence research*, *4*, 237-285.

Khan, A., & Salim, N. (2014). A review on abstractive summarization methods. *Journal of Theoretical and Applied Information Technology*, *59*(1), 64-72.

Knuth, D. E. (1973). *The art of computer programming, vol 3: Sorting and Searching*.

Kupiec, J., Pedersen, J., & Chen, F. (1995). A trainable document summarizer. In *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*, 68-73.

Landauer, T. K., & Dumais, S. T. (1997). A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review*, *104*(2), 211.

Lantz, B. (2019). *Machine learning with R: expert techniques for predictive modeling*. Packt publishing ltd., 18.

Le, Q., & Mikolov, T. (2014). Distributed representations of sentences and documents. In *International conference on machine learning*, 1188-1196.

Lin, C. Y. (2004). Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, 74-81.

Luhn, H. P. (1958). The automatic creation of literature abstracts. *IBM Journal of research and development*, *2*(2), 159-165.

Malkov, Y. A., & Yashunin, D. A. (2018). Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE transactions on pattern analysis and machine intelligence*, *42*(4), 824-836.

Mani, I., Maybury, M. (1999). *Advances in Automatic Text Summarization.* MIT Press.

Mihalcea, R., & Tarau, P. (2004). Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, 404-411.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mohri, M., Rostamizadeh, A., & Talwalkar, A. (2018). *Foundations of machine learning*. MIT press.

Muja, M., & Lowe, D. (2009). Flann-fast library for approximate nearest neighbors user manual. *Computer Science Department, University of British Columbia, Vancouver, BC, Canada*, *5*.

Papineni, K., Roukos, S., Ward, T., & Zhu, W. J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 311-318.

Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 1532-1543.

Ponomarenko, A., Malkov, Y., Logvinov, A., & Krylov, V. (2011). Approximate nearest neighbor search small world approach. In *International Conference on Information and Communication Technologies & Applications* (Vol. 17).

Saggion, H., & Lapalme, G. (2002). Generating indicative-informative summaries with sumum. *Computational linguistics*, *28*(4), 497-526.

Salton, G., & Buckley, C. (1988).Term-weighting approaches in automatic text retrieval. *Information processing & management, 24*(5), 513-523.

Samuel A.L. (1959). Some studies in machine learning using the game of checkers. *IBM Journal of research and development, 3*(3),210-29.

Stuart, J. (2010). Artificial Intelligence A Modern Approach Third Edition.

Sun, Y., Wang, W., Qin, J., Zhang, Y., & Lin, X. (2014). SRS: solving c-approximate nearest neighbor queries in high dimensional euclidean space with a tiny index. *Proceedings of the VLDB Endowment*.

Torun, H., & Inner, A. B. (2018). Detecting similar news by summarizing Turkish news. In *2018 26th Signal Processing and Communications Applications Conference (SIU)*, 1-4.

URL-1 https://github.com/spotify/annoy, (Ziyaret Tarihi: 12 Eylül 2021)

URL-2 https://github.com/FALCONN-LIB/FALCONN, (Ziyaret Tarihi: 20 Aralık 2021)

URL-3 https://en.wikipedia.org/wiki/Artificial_neural_network, (Ziyaret Tarihi: 10 Ağustos 2021)

URL-4 https://tr.wikipedia.org/wiki/Yapay_sinir_a%C4%9Flar%C4%B1, (Ziyaret Tarihi: 10 Ağustos 2021)

Uzundere, E., Dedja, E., Diri, B., & Amasyalı, M. F. (2008). Türkçe haber metinleri için otomatik özetleme. *Akıllı Sistemlerde Yenilikler ve Uygulamaları Sempozyumu,* 1-4.

Zhu, X. J. (2005). Semi-supervised learning literature survey.

KİŞİSEL YAYIN VE ESERLER

Torun, H., & Inner, A. B. (2018). Detecting similar news by summarizing Turkish news. In *2018 26th Signal Processing and Communications Applications Conference (SIU)*, 1-4.

Torun, H., & Inner, A. B. (2022). A Method for Similarity Detection in Vector Space by Summarizing News Articles. In *2022 30th Signal Processing and Communications Applications Conference (SIU)*.

ÖZGEÇMİŞ

Hakan Torun, ilk ve orta öğrenimini Erzurum'da, lise öğrenimini Yalova'da tamamladı. 2009 yılında başladığı Kocaeli Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümü'nden 2014 yılında mezun oldu. 2018-2022 yılları arasında, Kocaeli Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı'nda yüksek lisans eğitimini tamamladı. 2011 yılından beri çeşitli yazılım teknolojileri geliştirmekte ve XON Teknoloji A.Ş.'de Yönetim Kurulu Başkanı olarak görevini sürdürmektedir.